សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

មហាវិទ្យាល័យវិស្វកម្មកសិកម្ម

ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម

Design and Testing of Agricultural Machinery

រៀបរៀងដោយ

ឡោ លីតួ

ឧបត្ថម្ភដោយ

២០២១

សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

មហាវិទ្យាល័យវិស្វកម្មកសិកម្ម

ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម

Design and Testing of Agricultural Machinery

លោក ឡោ លីត្ឋ

២០២១

រក្សាសិទ្ធិ

©ឆ្នាំ២០២១

រក្សាសិទ្ធិគ្រប់យ៉ាង
គ្មានផ្នែកណាមួយនៃសៀវភៅនេះ អាចត្រូវបានថតចម្លង និងផលិតឡើងវិញ ដោយគ្មានការអនុញ្ញាត ជាលាយលក្ខណ៍អក្សរពីអ្នកនិពន្ធឡើយ។

បោះពុម្ពលើកទី១ ដោយមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិតនិងនវានុវត្តន៍ (ស.គ.ន) នៃក្រសួងអប់រំ យុវជន និងកីឡា នៅព្រះរាជាណាចក្រកម្ពុជា។

ទំនាក់ទំនងព័ត៌មាន ៖
អ្នកនិពន្ធ ៖ ឡោ លីតួ
ទូរស័ព្ទ ៖ (៨៥៥) ១១ ៩៧៣ ៦៧២
អ៊ីមែល ៖ llytour@rua.edu.kh

©2021. Lytour Lor All right reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any process without prior written permission from the author.

First Edition
Printed in the Kingdom of Cambodia

Enquiries about the book:
Author: Lytour Lor
Mobile phone: (855) 11 973 672
Email: llytour@rua.edu.kh

មុខកថា

ដំណើរអភិវឌ្ឍនៃព្រះរាជាណាចក្រកម្ពុជានៅក្នុងយុគសម័យទំនើបនេះ ជាមេរៀនដ៏ជោគជ័យបំផុតមួយ ដែលចាប់ឫសគល់ចេញពីការបញ្ចប់របបប្រល័យពូជសាសន៍ ការបញ្ចប់សង្គ្រាម ការផ្សះផ្សារជាតិ ការកសាងមូលដ្ឋានរឹងមាំនៃសន្តិភាពនិងស្ថេរភាព និងការអភិវឌ្ឍសេដ្ឋកិច្ច។ នៅក្រោយពេលដែលសន្តិភាពត្រូវបានកើតឡើងដោយបរិបូណ៌នៅឆ្នាំ១៩៩៨ កម្ពុជាទទួលកំណើនសេដ្ឋកិច្ចខ្ពស់ គឺប្រមាណ៨% ក្នុងមួយឆ្នាំ។ លើសពីនេះទៀត អត្រានៃភាពក្រីក្រត្រូវបានកាត់បន្ថយពីប្រមាណ៥៣% នៅឆ្នាំ២០០៤ មកនៅទាបជាង១០% នៅឆ្នាំ២០១៩។ ដំណើរនៃការអភិវឌ្ឍជាតិជាសកម្មភាពដែលបន្តទៅមុខជាប់ជានិច្ច ហើយគោលនយោបាយថ្មីៗដែលមានលក្ខណៈអន្តរវិស័យគ្រប់ជណ្តប់ក៏កំពុងលេចរូបរាងឡើង ដើម្បីតម្រង់ទិសកម្ពុជាឆ្ពោះទៅកាន់ប្រទេសមានប្រាក់ចំណូលមធ្យមកម្រិតខ្ពស់នៅឆ្នាំ២០៣០ និងឈានឡើងជាប្រទេសមានប្រាក់ចំណូលខ្ពស់ នៅឆ្នាំ២០៥០។ ការប្រែប្រួលឆាប់រហ័សនៃនិម្មាបនកម្មពិភពលោកនិងតំបន់ រួមទាំងទំនាក់ទំនងភូមិសាស្ត្រនយោបាយ បានផ្តល់កាលានុវត្តភាពសម្រាប់ការអភិវឌ្ឍឧស្សាហកម្មនៅកម្ពុជា ដែលត្រូវបានរាជរដ្ឋាភិបាលចាត់ទុកជាមូលដ្ឋានគ្រឹះនៃកំណើនសេដ្ឋកិច្ចកម្ពុជា។ រាជរដ្ឋាភិបាលកម្ពុជាបាន និងកំពុងបន្តពង្រឹងនិងអភិវឌ្ឍវិស័យអប់រំឆ្ពោះទៅរកការស្រាវជ្រាវនិងនវានុវត្តន៍ ដើម្បីពង្រឹងសមត្ថភាពនិងជំនាញរបស់ធនធានមនុស្សនៅកម្ពុជា ឱ្យស្របទៅនឹងបរិបទថ្មីនៃការអភិវឌ្ឍ ជាពិសេសការពង្រឹងសហគ្រិនភាពក្នុងការរៀបចំម៉ូដែលធុរកិច្ចថ្មីៗ។ ដើម្បីចាប់យកកាលានុវត្តភាពពីបដិវត្តន៍ឧស្សាហកម្មទី៤ និងសេដ្ឋកិច្ចឌីជីថលដែលកំពុងផុសផុលឡើង ប្រព័ន្ធអេកូឡូស៊ីដែលបង្កលក្ខណៈអំណោយផលដល់ការបង្កើតថ្មី នវានុវត្តន៍ ការស្រាវជ្រាវ និងអភិវឌ្ឍន៍ត្រូវតែមានការកែលម្អ។

បណ្តាប្រទេសនៅទ្វីបអាស៊ីកំពុងនាំមុខក្នុងការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ ដោយមានភាគហ៊ុនប្រមាណ៤៤% នៃការវិនិយោគទាំងមូលរបស់ពិភពលោក។ ប្រទេសចិនកំពុងបន្តកសាងហេដ្ឋារចនាសម្ព័ន្ធនៃការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ ក៏ដូចជាសមត្ថភាពមនុស្ស។ ផ្ទុយទៅវិញប្រទេសនៅទ្វីបអាមេរិកខាងត្បូងនិងអាហ្វ្រិក កំពុងស្ថិតនៅឆ្ងាយពីការវិនិយោគនេះ ហើយជាលទ្ធផលប្រទេសទាំងនោះក៏ពុំមានកំណើនសេដ្ឋកិច្ចគួរឱ្យកត់សម្គាល់ដែរ។ ទុនវិនិយោគសរុបលើការស្រាវជ្រាវនិងអភិវឌ្ឍរបស់ប្រទេសនៅទ្វីបអាមេរិកខាងត្បូនិងអាហ្វ្រិក មានប្រមាណ៥%នៃការវិនិយោគទាំងមូលរបស់ពិភពលោក ក្នុងពេលដែលតំបន់ទាំង២នេះមានប្រជាជនប្រមាណ២០%នៃប្រជាជនពិភពលោក។ ប្រទេសចំនួន៦ដែលមានលំដាប់ខ្ពស់ជាងគេនៅក្នុងការវិនិយោគលើការស្រាវជ្រាវនិងអភិវឌ្ឍ រួមមានសហរដ្ឋអាមេរិក ចិន ជប៉ុន អាល្លឺម៉ង់ ឥណ្ឌា និងកូរ៉េខាងត្បូង ដែលស្មើនឹងប្រមាណ៧០%នៃទុនវិនិយោគសរុបរបស់ពិភពលោក។

តើចំណេះដឹង ផលិតផល និងសេវាកម្មថ្មីទាំងនេះកើតឡើងពីអ្វី? ហើយកើតឡើងដោយរបៀបណា? ព្រះរាជាណាចក្រកម្ពុជាកំពុងតែកសាងមូលដ្ឋានសម្រាប់ការត្រៀមខ្លួនទទួល និងប្រកួតប្រជែងក្នុងយុគសម័យបដិវត្តឧស្សាហកម្មទី៤ នៅក្នុងសេដ្ឋកិច្ចដែលផ្អែកលើពុទ្ធិ ហើយដែលប្រការនេះ

ចាំបាច់តម្រូវឱ្យពលរដ្ឋកម្ពុជា ត្រូវក្លាយខ្លួនជាពលរដ្ឋឌីជីថល ពលរដ្ឋសកល និងពលរដ្ឋដែលប្រកបដោយការទទួលខុសត្រូវ ដែលមានសមត្ថភាពក្នុងការផលិត ចែកចាយ និងប្រើប្រាស់ពុទ្ធិដើម្បីទទួលមនុញ្ញផល និងរួមចំណែកក្នុងកំណើន។ ធនាគារពិភពលោកបានធ្វើការកត់សម្គាល់តាំងពីឆ្នាំ ២០០២នូវបម្លាស់ប្តូរនៃមូលដ្ឋានសេដ្ឋកិច្ច ពីសេដ្ឋកិច្ចដែលពឹងផ្អែកលើកម្លាំងពលកម្ម និងធនធានអតិកម្ម (Labour and Resource Based Economy) ទៅកាន់សេដ្ឋកិច្ចដែលពឹងផ្អែកលើពុទ្ធិ (Knowledge Based-Economy) ដែលក្នុងន័យនេះ ពុទ្ធិគឺជាគន្លឹះនៃការអភិវឌ្ឍ។ អាស្រ័យហេតុនេះ នៅលើគន្លងដែលកម្ពុជាកំពុងធ្វើដំណើរឆ្ពោះទៅកាន់សេដ្ឋកិច្ចឌីជីថល សង្គមកម្ពុជាត្រូវតែមានសមត្ថភាពក្នុងការផលិត ជ្រើសរើស បន្សាំ បង្កើតមុខរបរ និងប្រើប្រាស់ពុទ្ធិ ដើម្បីរក្សានិរន្តរភាពនៃកំណើន និងកែលម្អជីវភាពរស់នៅ។ សមត្ថភាពទាំងនេះ អាចកើតឡើងនៅពេលពលរដ្ឋកម្ពុជាមានឱកាសក្នុងការទទួលបានបទពិសោធន៍ពីការស្រាវជ្រាវ ការបណ្តុះគំនិតច្នៃប្រឌិត និងការស្វែងរកនវានុវត្តន៍។

កំណែទម្រង់វិស័យអប់រំ គឺជាការត្រួសត្រាយមាគ៌ាសម្រាប់ដំណើរឆ្ពោះទៅកាន់សង្គមប្រកបដោយពុទ្ធិ និងប្រជាពលរដ្ឋប្រកបដោយភាពរស់រវើក។ តាមរយៈមូលដ្ឋានអប់រំ សង្គមប្រកបដោយពុទ្ធិនឹងប្រមូលផ្តុំ បង្កើត និងចែករំលែក ទៅកាន់សមាជិកក្នុងសង្គមនូវសម្បទាអប់រំ ពិសេសគឺពុទ្ធិសម្បទាក្នុងបុព្វហេតុនៃមនុស្សជាតិនិងឧត្តមប្រយោជន៍នៃប្រទេស។ សង្គមប្រកបដោយពុទ្ធិ គឺពុំគ្រាន់តែជាសង្គមដែលសម្បូរព័ត៌មានប៉ុណ្ណោះទេ តែជាសង្គមដែលប្រជាពលរដ្ឋអាចធ្វើបរិវត្តកម្មពីព័ត៌មានទៅជាមូលធនប្រកបដោយប្រសិទ្ធភាព។ ការរីកចម្រើនទៅមុខជាលំដាប់នៃបច្ចេកវិទ្យានិងតំណភ្ជាប់ បានពង្រីកព្រំដែននៃការចូលទៅកាន់ និងការទទួលបានព័ត៌មានជាសកល ហើយដែលក្នុងន័យនេះ ការអប់រំនឹងបន្តវិវត្តទៅមុខនិងមានការផ្លាស់ប្តូរ។ សង្គមមួយដែលមានអំណាន និងរបាប់ជាបុរេលក្ខខណ្ឌនៃជីវភាពប្រចាំថ្ងៃនៃប្រជាពលរដ្ឋ ពេលនោះបំណិននៃអំណាន និពន្ធ និងការគណនាលេខនព្វន្ត គឺជាចលករនៃការរៀនរបស់សិស្ស។ ជាតុដ៏ចម្បងមួយដែលស្ថិតនៅក្នុងការកសាងសង្គមដែលប្រកបដោយពុទ្ធិគឺសៀវភៅសិក្សា ហើយការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សាជាប្រចាំ គឺជានវានុវត្តន៍នៃវិស័យអប់រំដែលនាំទៅរកការសិក្សាពេញមួយជីវិត ការអភិវឌ្ឍសម្បទាអប់រំ និងការចែករំលែកចំណេះដឹង។ មូលដ្ឋានអប់រំ ជាពិសេសគឺគ្រឹះស្ថានឧត្តមសិក្សាត្រូវមានតួនាទីដែលប្រកបដោយការឆ្លើយតប ចំពោះតម្រូវការខាងលើនេះ។ សាស្ត្រាចារ្យ អ្នកស្រាវជ្រាវ និងបុគ្គលិកអប់រំត្រូវបន្តសិក្សាជាប់ជានិច្ច តាមរយៈការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សា ហើយដែលសៀវភៅសិក្សាទាំងនេះនឹងក្លាយជាស្ពាននៃទំនាក់ទំនងរវាងនវានុវត្តន៍នៃបច្ចេកវិទ្យា និងការរៀននិងបង្រៀននៅក្នុងថ្នាក់រៀន។

សង្គមដែលប្រកបពុទ្ធិ ក៏ជាសង្គមដែលបណ្តុះឱ្យមានរចនាសម្ព័ន្ធទន់នៃសេដ្ឋកិច្ចដែលពឹងផ្អែកលើពុទ្ធិដែរ។ ឧទាហរណ៍ជាក់ស្តែងនៃបែបផែននេះរួមមាន Silicon Valley នៃសហរដ្ឋអាមេរិក សួនឧស្សាហកម្មវិស្វកម្មអាកាសយានយន្តនិងយានយន្តនៅទីក្រុង Munich ប្រទេសអាល្លឺម៉ង់ តំបន់ជីវបច្ចេកវិទ្យានៅក្រុង Hyderabad ប្រទេសឥណ្ឌា តំបន់ផលិតគ្រឿងអេឡិចត្រូនិកនិងសារគមនាគមន៍ឌីជីថលនៅទីក្រុង Seoul ប្រទេសកូរេខាងត្បូង ក៏ដូចជាសួនឧស្សាហកម្មថាមពល និងឥន្ធនគីមីសាស្ត្រនៃប្រទេសប្រេស៊ីល ហើយក៏នៅមានទីក្រុងនៃប្រទេសជាច្រើនទៀតនៅលើពិភពលោក។ លក្ខណៈសម្បត្តិ

នៃទីក្រុងទាំងនេះគឺការប្រើប្រាស់និន្នាការនៃការអភិវឌ្ឍដែលជំរុញ និងតម្រង់ទិសដោយចំណេះដឹង ហើយដែលចំណេះដឹងទាំងនោះកើតចេញជាដំបូងពីការវិនិយោគទៅលើគ្រឹះស្ថានឧត្តមសិក្សា ស្ថាប័នស្រាវជ្រាវ មជ្ឈមណ្ឌលឧត្តមភាពនៃជំនាញជាន់ខ្ពស់ ការប្រកួតប្រជែងដោយគុណាធិបតេយ្យ និងជាពិសេសគឺការបណ្តុះវប្បធម៌អំណាននិងនិពន្ធសៀវភៅ។ ល្បឿននៃការរីកចម្រើនផ្នែកពុទ្ធិ និងបច្ចេកវិទ្យាកំពុងមានសន្ទុះលឿនជាងអ្វីដែលសិស្ស និងនិស្សិតអាចទទួលបានពីគ្រូនៅគ្រឹះស្ថានសិក្សា ដែលធ្វើឱ្យគោលដៅនៃការអប់រំនៅពេលបច្ចុប្បន្ននេះ មានការប្រឈមខ្លាំងជាងពេលណាទាំងអស់។ ឧទាហរណ៍ក្នុងមួយឆ្នាំ មានសៀវភៅជាង២,២លានចំណងជើង ត្រូវបានសរសេរនិងបោះពុម្ព ដែលក្នុងនោះប្រទេសចិនមាន៤៤០ពាន់ ចំណែកឯសហរដ្ឋអាមេរិកមាន៣០៥ពាន់ និងប្រទេសរុស្ស៊ីមាន១២០ពាន់ចំណងជើង។

ខណៈពេលដែលបច្ចេកវិទ្យាកំពុងរីកចម្រើនជារៀងរាល់ថ្ងៃ មធ្យោបាយសម្រាប់អំណានក៏មានច្រើនជម្រើសសម្រាប់សិស្ស-និស្សិត និងសាធារណៈជន រួមមានការអានសៀវភៅ ការអានលើឧបករណ៍អេឡិចត្រូនិក ការអានដោយប្រើទូរសព្ទវៃឆ្លាត និងការអានលើកុំព្យូទ័រ ដែលសុទ្ធសឹងជាមធ្យោបាយសំខាន់ៗដែលនាំអ្នកអានទាំងឡាយឱ្យសម្រេចគោលបំណងអានរបស់ខ្លួន។ ម្យ៉ាងវិញទៀត អំណានដោយប្រើមធ្យោបាយបច្ចេកវិទ្យាទំនើប ចំណាយពេលតិច ងាយស្រួលអាន និងជួយដល់បរិស្ថានមួយកម្រិតទៀត។ នាពេលបច្ចុប្បន្ន សិស្ស-និស្សិត និងសាធារណៈជនកម្ពុជាដែលស្រឡាញ់អំណានកំពុងតែប្រើប្រាស់មធ្យោបាយអំណានទាំងនេះ។ បើយើងក្រឡេកមើលទៅប្រទេសជឿនលឿន ទោះបីជាបច្ចេកវិទ្យារីកចម្រើនខ្លាំងយ៉ាងណា អំណានតាមរយៈសៀវភៅនៅតែមានសន្ទុះដដែល។ ម្យ៉ាងវិញទៀត បច្ចេកវិទ្យាអានបែបទំនើបតាមរយៈឧបករណ៍ទំនើប អាស្រ័យលើលទ្ធភាពនៃធនធានអប់រំឌីជីថល និងមាតិកាឌីជីថលគ្រប់គ្រាន់ដែលបានផលិត និងបង្ហោះចែកចាយសម្រាប់អំណាន។

ក្នុងបរិបទកម្ពុជា ជាពិសេសក្នុងបរិការណ៍នៃការផ្ទុះរីករាលដាលនៃជំងឺកូវីដ-១៩ ក្រសួងអប់រំ យុវជន និងកីឡា បានជំរុញឱ្យមានបរិវត្តកម្មឌីជីថលនៅក្នុងអេកូស៊ីស្តែមនៃការអប់រំ ជាពិសេសការអប់រំតាមប្រព័ន្ធអេឡិចត្រូនិកនិងការអប់រំពីចម្ងាយ ដើម្បីលើកកម្ពស់អំណាន តាមរយៈការផលិតមាតិកាឌីជីថលដែលមានភាពចំរុះ ការកសាងសមត្ថភាពផ្នែកតំណភ្ជាប់និងវេទិកាឌីជីថល ការពង្រីកវិសាលភាពនៃមជ្ឈមណ្ឌលទិន្នន័យ និងការលើកកម្ពស់គុណភាពនៃការផលិតធនធានអប់រំឌីជីថល គួបផ្សំជាមួយការចែកសន្លឹកកិច្ចការឱ្យសិស្សយកទៅរៀននៅផ្ទះ និងការចុះទៅជួបជាមួយសិស្សជាបណ្តុំនៅតាមសហគមន៍។ ក្នុងន័យលើកកម្ពស់អំណាន និងភាពសម្បូរបែបនៃធនធានសៀវភៅសិក្សា ឱ្យកាន់តែមានប្រសិទ្ធភាពនិងភាពសក្តិសិទ្ធិ និងផ្តល់ឱកាសអំណានកាន់តែច្រើនថែមទៀតដល់សិស្សានុសិស្ស និស្សិត និងសាធារណៈជន ក្រសួងអប់រំ យុវជន និងកីឡាលើកទឹកចិត្តនូវចំណុចមួយចំនួនដូចខាងក្រោម៖

១. សាស្ត្រាចារ្យ អ្នកស្រាវជ្រាវ និងបុគ្គលិកអប់រំ សូមបន្តនិងបង្កើនការបោះពុម្ពស្នាដៃបន្ថែមទៀត ដើម្បីធ្វើឱ្យធនធានសម្រាប់អំណានកាន់តែសម្បូរបែប ជាពិសេសធនធានអំណានជាខេមរភាសា

២. គ្រឹះស្ថានឧត្តមសិក្សា សូមផ្តល់លទ្ធភាពគ្រប់បែបយ៉ាង ដើម្បីឱ្យបុគ្គលិកអប់រំគ្រប់លំដាប់ថ្នាក់ និងនិស្សិតគ្រប់កម្រិតសិក្សាអាចចូលរួមអាន និងសិក្សាស្រាវជ្រាវតាមគ្រប់លទ្ធភាពជាមួយធនធានអំណាន ជាពិសេសការរៀបចំឱ្យមានពេលវេលាសម្រាប់សហសិក្សា និងអំណានក្នុងបណ្ណាល័យ
៣. សាស្ត្រាចារ្យតាមមុខវិជ្ជា និងអ្នកស្រាវជ្រាវតាមជំនាញឬវិស័យ ត្រូវរៀបចំដំណើរការរៀនបង្រៀន និងស្រាវជ្រាវដែលមានដាក់បញ្ចូលកិច្ចការស្វ័យសិក្សា សហសិក្សា ឬការស្រាវជ្រាវបណ្ណាល័យដែលតម្រូវឱ្យនិស្សិត ត្រូវអាននិងស្រាវជ្រាវជាមួយធនធានអំណាន
៤. គ្រឹះស្ថានឧត្តមសិក្សា និងមជ្ឈមណ្ឌលស្រាវជ្រាវ ត្រូវខិតខំឱ្យអស់លទ្ធភាពក្នុងការបង្កើតបណ្ណាល័យ មជ្ឈមណ្ឌលរក្សាឯកសារ ឬមជ្ឈមណ្ឌលអប់រំឌីជីថលជាដើម ដើម្បីឱ្យបុគ្គលិកអប់រំគ្រប់លំដាប់ថ្នាក់និងនិស្សិតគ្រប់កម្រិតសិក្សាអាចទទួលបាន និងស្វែងរកប្រភពសម្រាប់អំណានកាន់តែសម្បូរបែប និងមានភាពបត់បែន ឆ្លើយតបតាមតម្រូវការអ្នកអាន
៥. និស្សិតគ្រប់កម្រិតសិក្សាត្រូវខិតខំនិងចំណាយពេលវេលាដើម្បីអាន និងចាត់ទុកវប្បធម៌និងអកប្បកិរិយាអំណានជាផ្នែកមួយ នៃពេលវេលានិងភាពស៊ីវិល័យនៃជីវិតប្រចាំថ្ងៃ
៦. បងប្អូនជនរួមជាតិ ដែលជាមាតាបិតា ឬអ្នកអាណាព្យាបាល សូមជួយជំរុញនិងបង្កលក្ខណៈកាន់តែច្រើនថែមទៀត ជាពិសេសការលៃលកចំណាយនៅក្នុងគ្រួសារសម្រាប់ការទិញសម្ភារៈសិក្សា សៀវភៅអាន និងឧបករណ៍សម្រាប់អំណានដល់កូនៗ ដែលចាត់ទុកជាការវិនិយោគមួយដ៏សំខាន់ សម្រាប់ បង្កើនចំណេះដឹង និងអនាគតរបស់ពួកគេ។

ដោយមានការគាំទ្រពីក្រសួងសេដ្ឋកិច្ច និងហិរញ្ញវត្ថុ នៅឆ្នាំ២០២០ ក្រសួងអប់រំ យុវជន និងកីឡា បានបង្កើតមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ ដែលហៅកាត់ថា “មូលនិធិ ស.គ.ន.” និងហៅជាភាសាអង់គ្លេសថា The Research Creativity and Innovation Fund ដែលហៅកាត់ជាភាសាអង់គ្លេសថា “RCI Fund”។ គោលដៅចម្បងនៃមូលនិធិនេះ គឺរួមចំណែកលើកកម្ពស់វប្បធម៌នៃការស្រាវជ្រាវ បំផុសគំនិតច្នៃប្រឌិត និងជំរុញការធ្វើនវានុវត្ត ដើម្បីជាប្រយោជន៍ដល់វិស័យអប់រំ យុវជន និងកីឡា ដែលឆ្លើយតបទៅនឹងទីផ្សារពលកម្ម និងសាកលភាវូបនីយកម្ម។ មូលនិធិ ស.គ.ន. បានសម្រេចកំណត់ប្រធានបទ ជាអាទិភាពសម្រាប់ការគាំទ្រដោយមូលនិធិចំនួន៣ រួមមានឌីជីថលនីយកម្មសម្រាប់បដិវត្តឧស្សាហកម្ម៤.0 (Digitalization for IR.4.0) ការស្រាវជ្រាវអនុវត្តលើវិស័យកសិកម្ម (Applied Agricultural Research) និងការស្រាវជ្រាវគរុកោសល្យសតវត្សទី២១ (21st Century Pedagogy Research)។

ដោយមានការធ្វើអាទិភាពូបនីយកម្មទៅលើទិសដៅ នៃការប្រើប្រាស់ថវិកាមូលនិធិសម្រាប់ឆ្នាំ ២០២០ ក្រសួងសេដ្ឋកិច្ច និងហិរញ្ញវត្ថុ និងក្រសួងអប់រំ យុវជន និងកីឡា បានផ្តល់ការគាំទ្រដល់ការរៀបរៀង និពន្ធ និងកែលម្អ សៀវភៅសិក្សា (Text book) ដែលនឹងត្រូវប្រើប្រាស់នៅកម្រិតឧត្តមសិក្សា។ គោលបំណងនៃការរៀបរៀង និពន្ធ និងកែលម្អ សៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា គឺដើម្បីបង្កើនបរិមាណ លើកកម្ពស់គុណភាព និងពង្រីកសមធម៌នៃធនធានសិក្សាជាខេមរភាសា ជូនដល់និស្សិត

ដែលកំពុងបន្តការសិក្សា និងត្រៀមខ្លួនធ្វើការស្រាវជ្រាវនៅកម្រិតឧត្តមសិក្សា។ លើសពីនេះទៀត ការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា មានគោលដៅដូចខាងក្រោម ៖

១. ឆ្លើយតបជាបន្ទាន់ចំពោះការខ្វះខាតធនធានសិក្សា ដែលជាតម្រូវការសិក្សារបស់និស្សិតនៅកម្រិតឧត្តមសិក្សា
២. លើកកម្ពស់ទំនើបភាវូបនីយកម្ម និងឧត្តមានុវត្តន៍នៃការរៀននិងបង្រៀន និងការស្រាវជ្រាវនៅលើមុខវិជ្ជា កម្មវិធីសិក្សា ឬមុខជំនាញជាក់លាក់
៣. បង្កើនភាពស៊ីជម្រៅក្នុងការកសាងវិជ្ជាជីវៈនិងបទពិសោធន៍សម្រាប់បានៈសាស្ត្រាចារ្យ និងអ្នកស្រាវជ្រាវ
៤. រួមចំណែកដល់ការកសាងភាពជាសហគមន៍វិជ្ជាជីវៈ ការចែករំលែកបទពិសោធន៍ និងវប្បធម៌នៃការរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា។

ក្រសួងអប់រំ យុវជន និងកីឡា បានវាយតម្លៃខ្ពស់ចំពោះការបោះជំហានប្រកបដោយមនសិការវិជ្ជាជីវៈនៃគ្រឹះស្ថានឧត្តមសិក្សា និងបុគ្គលិកអប់រំទាំងអស់ ក្នុងការរៀបចំ រៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សា ដើម្បីបង្កើនបរិមាណ លើកកម្ពស់គុណភាព និងពង្រឹងសមធម៌នៃធនធានសិក្សាជាខេមរភាសា ជូននិស្សិតដែលកំពុងបន្តការសិក្សា និងត្រៀមខ្លួនធ្វើការស្រាវជ្រាវនៅកម្រិតឧត្តមសិក្សា។ សៀវភៅសិក្សាជាផ្នែកមួយនៃការទទួលស្គាល់គុណភាពអប់រំនៃគ្រឹះស្ថានឧត្តមសិក្សា និងជាធនធានសិក្សាដែលជាមូលដ្ឋានមួយដ៏សំខាន់ ក្នុងការគាំទ្រដល់ការបង្រៀន និងរៀន ហើយត្រូវមានបរិមាណគ្រប់គ្រាន់ ឆ្លើយតបទៅនឹងកម្មវិធីអប់រំ និងតម្រូវការសិក្សាស្រាវជ្រាវ។ ជាគោលការណ៍ គ្រឹះស្ថានឧត្តមសិក្សាទាំងអស់ ត្រូវមានសៀវភៅសិក្សាដែលប្រើជាគោលសម្រាប់មុខវិជ្ជានីមួយៗ។ ចំនួនសៀវភៅសិក្សាដែលគ្រប់គ្រាន់សម្រាប់ការស្រាវជ្រាវ និងការសិក្សារបស់និស្សិត ត្រូវមានយ៉ាងតិចមួយចំណងជើងក្នុងមួយមុខវិជ្ជា ហើយត្រូវតម្កល់យ៉ាងតិច២ច្បាប់នៅក្នុងបណ្ណាល័យ ឬអាចរកបានតាមប្រព័ន្ធអេឡិចត្រូនិក។ ក្រសួងអប់រំ យុវជន និងកីឡា លើកទឹកចិត្តបន្ថែមទៀតជូនដល់គ្រឹះស្ថានឧត្តមសិក្សារដ្ឋ និងឯកជនដែលបានស្នើសុំថវិកាមូលនិធិ ស.គ.ន រួច សូមចូលរួមបន្ថែមទៀតដើម្បីបង្កើនចំនួនចំណងជើងសៀវភៅ។ ចំណែកគ្រឹះស្ថានឧត្តមសិក្សារដ្ឋ និងឯកជនដែលពុំទាន់បានដាក់ពាក្យស្នើសុំថវិកាមូលនិធិ ដើម្បីរៀបរៀង និពន្ធ និងកែលម្អ សៀវភៅសិក្សានៅកម្រិតឧត្តមសិក្សា សូមរួសរាន់ចូលរួមដើម្បីជាគុណប្រយោជន៍ដល់តម្រូវការដ៏ទទូចនិងថ្លៃថ្លានៃនិស្សិតកម្ពុជាក្នុងការសិក្សា និងស្រាវជ្រាវនៅកម្រិតឧត្តមសិក្សា។

សេចក្តីបញ្ជាក់
នៃមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍

សៀវភៅសិក្សានេះជាលទ្ធផលនៃការស្នើសុំអនុវត្តថវិកាមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ ក្នុងគម្រោងរៀបរៀង និពន្ធ និងកែលម្អសៀវភៅសិក្សា ដែលនឹងត្រូវប្រើប្រាស់នៅកម្រិតឧត្តមសិក្សា។ សៀវភៅសិក្សានេះ ត្រូវបានរៀបរៀង និពន្ធ ឬកែលម្អដោយមានការធានាអះអាងថាជាស្នាដៃរបស់អ្នកនិពន្ធផ្ទាល់ និងបានឆ្លងកាត់ត្រួតពិនិត្យ ផ្តល់យោបល់ និងវាយតម្លៃដោយក្រុមប្រឹក្សាអប់រំ ក្រុមប្រឹក្សាស្រាវជ្រាវ ឬក្រុមប្រឹក្សាដែលមានតម្លៃស្មើនៃគ្រឹះស្ថានឧត្តមសិក្សា និងតាមរយៈកិច្ចសន្យាដែលបានធ្វើឡើង និងដែលបានតម្កល់ទុកនៅមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍។ រាល់ខ្លឹមសារ ការបកស្រាយ ឬរូបភាព ដែលមាននៅក្នុងសៀវភៅនេះ គឺជាជំហរនិងទស្សនៈផ្ទាល់របស់អ្នកនិពន្ធ ហើយពុំឆ្លុះបញ្ចាំង ឬជាតំណាងដល់មូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ នៃក្រសួងអប់រំ យុវជន និងកីឡាឡើយ។

សេចក្ដីថ្លែងអំណរគុណ

ជាបឋម អ្នកនិពន្ធសូមថ្លែងអំណរគុណដ៏ជ្រាលជ្រៅជូនដល់**មូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍** (ស.គ.ន) **នៃក្រសួងអប់រំ យុវជន និងកីឡា** ដែលបានឧបត្ថម្ភថវិកាសម្រាប់ការតាក់តែង និងត្រួតពិនិត្យសៀវភៅនេះ ប្រសិនបើគ្មានការឧបត្ថម្ភនេះទេ សៀវភៅបច្ចេកទេសនេះនឹងមិនអាចលេចចេញជារូបរាង្គ និងសម្រាប់បម្រើដល់ការសិក្សាស្រាវជ្រាវរបស់និស្សិតនិងអ្នកពាក់ព័ន្ធបានឡើយ។

អ្នកនិពន្ធក៏សូមថ្លែងអំណរគុណផងដែរ ជូនដល់ឯកឧត្តមសាស្ត្រាចារ្យ **ង៉ោ ប៊ុនថាន** ដែលបានជួយត្រួតពិនិត្យ និងផ្ដល់យោបល់លើមាតិកា និងអត្ថន័យនៅក្នុងសៀវភៅនេះ។

សូមថ្លែងអំណរគុណដល់លោក **គីម ម៉េងឆាយ** ដែលបានជួយត្រួតពិនិត្យលើការរៀបចំ សរសេរសៀវភៅនេះមុននឹងបោះពុម្ព។

ជាចុងក្រោយ សូមថ្លែងអំណរគុណដល់ក្រុមលេខាធិការ នៃគណៈកម្មការតាក់តែងសៀវភៅ នៃសាកលវិទ្យាល័យភូមិន្ទកសិកម្ម ដែលបានកែលម្អការរចនាសៀវភៅឲ្យត្រូវទៅនឹងការណែនាំរបស់មូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ (ស.គ.ន) នៃក្រសួងអប់រំ យុវជន និងកីឡា។

អ្នកនិពន្ធ

គោត្តនាម និងនាម ៖ ឡេង លីតួ
អាសយដ្ឋាន ៖ ភូមិម័ល សង្កាត់ដង្កោ ខណ្ឌដង្កោ ភ្នំពេញ
ស្ថាប័នការងារ ៖ សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម
ឯកទេស ឬមុខជំនាញ ៖ វិស្វកម្មកសិកម្ម
ប្រវត្តិការសិក្សា ៖ ២០០២-២០០៣ ថ្នាក់បរិញ្ញាជាន់ខ្ពស់ វិស្វកម្មកសិកម្ម វិទ្យាស្ថានបច្ចេកវិទ្យាអាស៊ី
១៩៩៦-២០០១ ថ្នាក់បរិញ្ញា វិស្វកម្មកសិកម្ម សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

បទពិសោធន៍ការងារ ៖

-ចាប់ពីឆ្នាំ២០១៧ ជាព្រឹទ្ធបុរស មហាវិទ្យាល័យវិស្វកម្មកសិកម្ម សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

-ចាប់ពីឆ្នាំ២០១៤ ដល់ ២០១៧ ជាព្រឹទ្ធបុរសរងទទួលបន្ទុករួមមហាវិទ្យាល័យវិស្វកម្មកសិកម្ម

-ចាប់ពីឆ្នាំ២០០៤ ដល់ ២០១៤ ជានាយករងសាលាក្រោយឧត្តម នៃសាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

-ចាប់ពីឆ្នាំ២០១៦ ជាអ្នកសម្របសម្រួលមជ្ឈមណ្ឌលបច្ចេកវិទ្យានិងព័ត៌មានជីវឧស្ម័នខ្នាតធំ នៃសាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

-បង្រៀនមុខវិជ្ជា ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម នៅមហាវិទ្យាល័យវិស្វកម្មកសិកម្ម នៃសាកលវិទ្យាល័យភូមិន្ទកសិកម្ម ចាប់តាំងពីឆ្នាំ២០១៥

អារម្ភកថា

សៀវភៅស្តីពីការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្មនេះ ត្រូវបានតាក់តែងឡើងដើម្បីជាឯកសារយោងក្នុងមុខវិជ្ជា ម៉ាស៊ីនកសិកម្ម IV (ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម) សម្រាប់និស្សិតថ្នាក់បរិញ្ញាបត្រ នៃមហាវិទ្យាល័យវិស្វកម្មកសិកម្ម។ សៀវភៅនេះផ្តល់ចំណេះដឹងទាំងទ្រឹស្តី និងជំនាញដល់និស្សិត ដោយនៅក្នុងជំពូកទី១ ផ្តោតលើលើការរចនាប្លង់ និងការធ្វើតេស្តម៉ាស៊ីនកសិកម្ម ជំពូកទី២ ប្រព័ន្ធវាស់វែង កត់ត្រាទិន្នន័យ និងបញ្ជាស្វ័យប្រវត្តិ និងជំពូកទី៣ ម៉ាយក្រូកុនត្រូល័រ (Microcontroller) និងការប្រើប្រាស់ក្នុងការកត់ត្រាទិន្នន័យនិងស្វ័យប្រវត្តិកម្ម។

ការរចនាប្លង់ឧបករណ៍និងម៉ាស៊ីនកសិកម្ម និងផលិតផលផ្សេងៗទៀតដែលស្របនឹងតម្រូវការរបស់អ្នកប្រើប្រាស់ជាពិសេស ម៉ាស៊ីនប្រភេទ Smart ពិតជាចាំបាច់សម្រាប់សម័យកាលបច្ចុប្បន្នដែលកម្ពុជាផ្តោតលើការអភិវឌ្ឍឧស្សាហកម្ម ៤.០។ ម្យ៉ាងទៀត ការប្រើប្រាស់ប្រព័ន្ធស្វ័យប្រវត្តិសម្រាប់ប្រមូលនិងកត់ត្រាទិន្នន័យ ជាពិសេសប្រើប្រាស់ microcontroller និង sensor ក្នុងការធ្វើតេស្ត prototype ក៏ដូចសម្រាប់ស្វ័យប្រវត្តិកម្មក្នុងវិស័យកសិកម្មដើម្បីកាត់បន្ថយតម្រូវការកំលាំងពលកម្មនិងបង្កើនផលិតភាពការងារ ជាចំណេះដឹង និងជំនាញ ដែលវិស្វករ និងអ្នកបច្ចេកទេសវិស្វកម្មកសិកម្មត្រូវមានក្នុងយុគសម័យថ្មីនេះ។

សៀវភៅនេះរួមបញ្ចូលចំណេះដឹងគ្រឹះ និងឧទាហរណ៍ ពាក់ព័ន្ធនឹងវិស័យកសិកម្ម ដែលឧស្សាហ៍ប្រើប្រាស់ ប៉ុន្តែចំណេះដឹងនេះក៏ជាគ្រឹះសម្រាប់ប្រើប្រាស់ក្នុងវិស័យឧស្សាហកម្មផងដែរ។ ជាចុងក្រោយ ដោយសារសៀវភៅនេះជាសៀវភៅនិពន្ធលើកដំបូង ប្រាកដជានៅមានចំណុចខ្វះខាត ហេតុនេះ ខ្ញុំបាទរីករាយនឹងទទួលការរិះគន់ស្ថាបនាពីប្រិយមិត្តអ្នកអានទាំងអស់ ដើម្បីកែលម្អសៀវភៅនេះឲ្យកាន់តែប្រសើរនិងមានលក្ខណៈគ្រប់ជ្រុងជ្រោយបន្ថែមទៀត។

សូមអរគុណ

មាតិកា

ទំព័រ

បុព្វកថា និងសេចក្តីបញ្ជាក់នៃមូលនិធិ

សេចក្តីថ្លែងអំណរគុណ...i

អ្នកនិពន្ធ..ii

អារម្ភកថា...ii

មាតិកា...iv

ជំពូកទី១
ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម

១.១ ការរចនាប្លង់ ១

១.១.១ ទស្សនវិជ្ជានៃការរចនាប្លង់ ១

១.១.២ ដំណើរការរចនាប្លង់ ១

១.១.៣ ប្រភេទផ្សេងៗនៃដំណើរការរចនាប្លង់ ៤

១.១.៤ Human-centered Design ៤

១.២ ឧបករណ៍សម្រាប់ប្រមូលតម្រូវការរបស់អ្នកប្រើប្រាស់.. ៥

១.២.១ វិធីជំរុញឲ្យអ្នកចូលរួមបញ្ចេញគំនិត ៥

១.២.២ ច្បាប់សំខាន់ៗដែលត្រូវគោរពតាម.. ៦

១.២.៣ ប្រភេទឧបករណ៍ប្រើប្រាស់ ៦

១.២.៣.១ ព្យុះគំនិតប្រើប្រាស់ផ្ទាំងក្រដាសធំ (Flip Chart Brainstorming) ៦

១.២.៣.២ ព្យុះគំនិតប្រើប្រាស់សន្លឹកក្រដាសតូចៗ (Post-it Brainstorming)...................... ៧

១.២.៥ វិធីជួយឲ្យអ្នកចូលរួមមានគំនិត.. ៧

១.៣ ការធ្វើតេស្តម៉ាស៊ីននិងឧបករណ៍កសិកម្ម.. ៨

១.៣.១ គោលបំណងនៃការធ្វើតេស្ត ៨

១.៣.២ Test Code... ៨

១.៣.៣ ប៉ារ៉ាម៉ែត្រដែលត្រូវវាស់វែង.. ៨

១.៣.៤ អ្នកប្រើប្រាស់របាយការណ៍ធ្វើតេស្ត..៨

ជំពូកទី២

ប្រព័ន្ធវាស់វែង កត់ត្រាទិន្នន័យ និងបញ្ជាស្វ័យប្រវត្តិ

២.១ ជាតុនៃគ្រឿងអេឡិកត្រូនិក ១០
២.១.១ រ៉េស៊ីស្តង់.. ១០
២.១.២ ឌីយ៉ូត ១១
២.១.៣ កុងដង់សាទ័រ ១២
២.១.៤ ត្រង់ស៊ីស្ទ័រ.. ១៤
២.១.៥ គ្រឿងអេឡិកត្រូនិកប្រើពន្លឺ ១៥
២.២ ការវាស់វែងនិងកត់ត្រាទិន្នន័យ.. ១៧
២.២.១.១ RTD (Resistance Temperature Detector).. ១៩
២.២.១.២ ទែរមីស្ទ័រ (Thermistors)..២០
២.២.១.៣ Thermocouples ២០
២.២.១.៤ សេនស័រសីតុណ្ហភាពប្រភេទ Solid State ២១
២.២.១.៥ សេនស័រសម្ពាធ...២១
២.២.១.៦ កុងតាក់ (Switches).. ២២
២.២.១.៧ Potentiometers ២២
២.២.១.៨ សេនស័រ Photoelectric ២២
២.២.១.៩ សេនស័រលំហូរ ២៤
២.២.២ សៀគ្វីកែសម្រួលលក្ខខណ្ឌ (Signal Conditioner) ២៤
២.២.៣ អំភ្លី (amplifier)..២៤
២.២.៤ ឧបករណ៍កត់ត្រា...២៥
២.២.៥ ដំណើរការទិន្នន័យ (Data Processor) ២៦
២.២.៦ អ្នកត្រួតពិនិត្យដំណើរការ (Process Controller) ២៧
២.៣ ការវិភាគវិស្វកម្ម (Engineering Analysis).. ២៧
២.៤ ប្រព័ន្ធបញ្ជាស្វ័យប្រវត្តិ...២៨

២.៤.១ ឧបករណ៍សម្រាប់ត្រួតពិនិត្យដំណើរការ ២៨
២.៤.១.១ ប្រព័ន្ធបញ្ជា Open-loop និង Closed-loop ២៩
២.៤.២ ឧបករណ៍សម្រាប់កែសម្រួលប៉ារ៉ាម៉ែត្រ ៣០
២.៤.២.១ ឧបករណ៍បញ្ចេញសកម្មភាព (Actuator) ៣០
២.៥ ប្រព័ន្ធកត់ត្រាទិន្នន័យ (Data Acquisition system) ៤១

ជំពូកទី៣
ម៉ាយក្រូកុនត្រូល័រ (Microcontroller) និងការប្រើប្រាស់ក្នុងការកត់ត្រាទិន្នន័យនិងស្វ័យប្រវត្តិកម្ម

៣.១ ការណែនាំពីម៉ាយក្រូកុនត្រូល័រ Arduino ៤២
៣.១.១ ប្រភេទផ្សេងៗនៃ Arduino ៤២
៣.១.២ សេនស័ររបស់ Arduino ៤៧
៣.១.៣ Arduino IDE ៤៨
៣.១.៤ Arduino Shields ៥៣
៣.១.៥ Serial Communication ៥៤
៣.២ ការសរសេរកម្មវិធីសម្រាប់បញ្ជា Arduino ៥៤
៣.២.១ ការសរសេរកម្មវិធីប្រើប្រាស់ Arduino IDE ៥៥
៣.២.២ ការសរសេរកម្មវិធី Arduino ប្រើប្រាស់ TinkerCAD ៥៧
៣.២.៣ ប្រភេទទិន្នន័យ (Data Type) ភាសាកុំព្យូទ័រ C របស់ Arduino ៦៤
៣.២.៤ ការសម្រេចចិត្តក្នុងភាសាកុំព្យូទ័រ C ៦៤
៣.២.៥ ការធ្វើឲ្យកម្មវិធីដំណើរការសារចុះឡើង ៧៦
៣.២.៦ Function នៅក្នុងភាសាកុំព្យូទ័រ C ៨៤
៣.៣ សេនស័រសីតុណ្ហភាពនិងសំណើមបរិយាកាស ៨៩
៣.៤ ការបង្ហាញព័ត៌មាននៅលើអេក្រង់ LCD ៩៥
៣.៤.១ ប្រភេទអេក្រង់ LCD ៩៥
៣.៤.២ Pin របស់អេក្រង់ LCD ៩៥
៣.៤.៣ របៀបតភ្ជាប់អេក្រង់ LCD ទៅ Arduino ៩៦
៣.៤.៤ ការបង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសនៅលើអេក្រង់ LCD ១០០

៣.៤.៥ ការបញ្ជាម៉ូទ័រអាស្រ័យលើសីតុណ្ហភាពនិងសំណើមបរិយាកាស...............................១០១
៣.៤.៦ Real-time Clock Module (DS3231)...១០៤
៣.៤.៧ ការកាត់ត្រាទិន្នន័យទៅក្នុង SD Card... ១០៧
៣.៥ ការសរសេរកម្មវិធីសម្រាប់ Arduino ក្នុងបញ្ជាឧបករណ៍បញ្ចេញសកម្មភាព..................... ១១៣
៣.៥.១ Arduino Relay Module ១១៣
៣.៥.២ Optocoupler/Opto-isolator..១១៥
៣.៥.៣ Variable Frequency Drive (VFD)..១១៥
៣.៥.៤ បញ្ជាម៉ូទ័រពាហ្សប្រើប្រាស់ Contactor... ១១៧
៣.៦ ការបញ្ជា Arduino ពីចម្ងាយតាមរយៈប្រព័ន្ធអ៊ីនធ័រណេត ១១៩
៣.៦.១ ការណែនាំពីប្រព័ន្ធអ៊ីនធ័រណេត...១១៩
៣.៦.២ NodeMCU ១២០
៣.៦.៣ បន្ថែម NodeMCU Board... ១២០
៣.៧ ការវាស់វែងឧស្ម័នដោយប្រើប្រាស់ Arduino ១៤១

សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម

ជំពូកទី១
ការរចនាប្លង់និងធ្វើតេស្តម៉ាស៊ីនកសិកម្ម

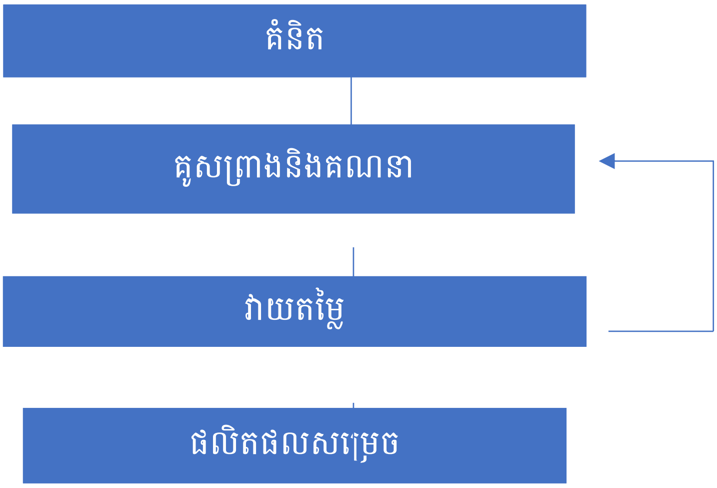
១.១ ការរចនាប្លង់

១.១.១ ទស្សនវិជ្ជានៃការរចនាប្លង់

សមត្ថភាពក្នុងការបង្កើតទស្សនវិជ្ជានៃការរចនាប្លង់ផ្ទាល់ខ្លួនជាគុណភាពរបស់អ្នករចនាប្លង់ដែលល្អ។ វារួមបញ្ចូលការរៀបចំវិធីសាស្ត្រចាំបាច់នៅក្នុងដំណើរការរចនាប្លង់ដើម្បីឈានពីគំនិត ទៅដល់បង្កើតផលិតផលសម្រេច ការកំណត់ធនធានខាងបច្ចេកទេស ដូចជាទំនាក់ទំនងក្នុងទម្រង់គណិតវិទ្យាដែលត្រូវប្រើក្នុងការគណនា ការគិតគូរខាងផ្នែកសេដ្ឋកិច្ច ធ្វើទាន់ពេលវេលា និងផលិតផលអាចជឿជាក់បាន (មានគុណភាពល្អ) និងសុវត្ថិភាព។ វិធីសាស្ត្រនិងទស្សនវិជ្ជានៃការរចនាប្លង់ផ្លាស់ប្តូរតាមការផ្លាស់ប្តូរបច្ចេកវិទ្យា ឧទាហរណ៍ ការប្រើប្រាស់កុំព្យូទ័រក្នុងការរចនាប្លង់ ជួយឲ្យការងារធ្វើបានលឿន និងកាត់បន្ថយចំណាយ។ ការសន្មត់និងការគណនាបន្ទុកសំខាន់ៗ ប្រសើរជាងការធ្វើដោយមិនបានគណនាស្មានៗជាមុន។ ម្យ៉ាងទៀត បទពិសោធន៍ក៏អាចជំនួសឲ្យការគណនាបានខ្លះដែរ ឧទាហរណ៍ គេអាចប្រើប្រាស់របស់ស្រដៀងដោយសារកសិករ ឬ ជាង ធ្លាប់ប្រើដូច្នេះបានមិនមានបញ្ហា។ វិស្វករវ័យក្មេងអាចមិនបន្តធ្វើប្រសិនបើមិនមានព័ត៌មានគ្រប់គ្រាន់ ប៉ុន្តែផ្ទុយទៅវិញជាងដែលទទួលបានការបណ្តុះបណ្តាលនិងមានបទពិសោធន៍អាចបន្តកែសម្រួលនៅពេលជួបបញ្ហារហូតដល់ចេញជាផលិតផលអាចប្រើប្រាស់បាន។

១.១.២ ដំណើរការរចនាប្លង់

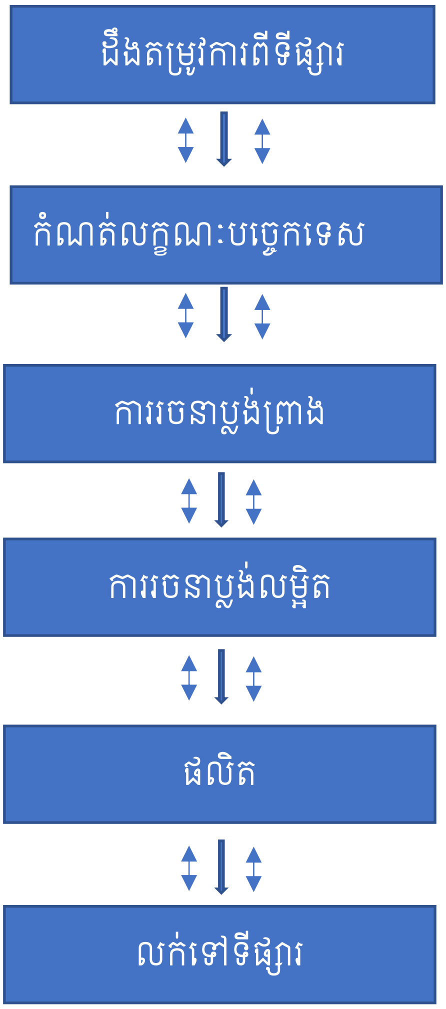
ដំណើរការរចនាប្លង់សាមញ្ញបង្ហាញនៅក្នុងរូបភាព១.១។



រូបភាព១.១៖ ដំណើរការរចនាប្លង់សាមញ្ញ

គំនិតក្នុងការរចនាប្លង់ថ្មីចាប់ផ្តើមចេញពីតម្រូវការរបស់អ្នកប្រើប្រាស់ ដើម្បីដោះស្រាយបញ្ហាអ្វីមួយ ឧទាហរណ៍ ឧបករណ៍បញ្ជាសម្រាប់ស្វ័យប្រវត្តិកម្មនៅក្នុងផ្ទះសំណាញ់បង្កើតឡើងដើម្បីកាត់បន្ថយការចំណាយកំលាំងពលកម្មក្នុងការមើលថែទាំដំណាំ។ ចេញពីគំនិតនេះ អ្នករចនាប្លង់ គិតនិងគូសព្រាងផ្នែក សំខាន់ៗរបស់ឧបករណ៍នេះ និងគិតពីទំនាក់ទំនងរបស់វាពីមួយទៅមួយ រួមទាំងការគណនាផ្សេងៗដើម្បីកំណត់ទំហំ និងលក្ខណៈបច្ចេកទេសរបស់ផ្នែកនីមួយៗ និងផលិតចេញជា prototype នៃឧបករណ៍នេះ។ គេត្រូវវាយតម្លៃ prototype តាមរយៈការធ្វើតេស្តដើម្បីដឹងថា ឧបករណ៍នេះអាចដោះស្រាយបញ្ហា និងត្រូវនឹងតម្រូវការរបស់អ្នកប្រើប្រាស់ដែរឬទេ? ប្រសិនបើ នៅមានចំណុចខ្វះខាត គេត្រូវគូសព្រាងនិងគណនាឡើង វិញ។ ប្រសិនបើ ឧបករណ៍ដែលកែសម្រួលហើយត្រូវនឹងតម្រូវការ គេអាចផលិតជាផលិតផលសម្រេចបាន។

ដោយសារម៉ាស៊ីន ឬឧបករណ៍សម័យថ្មី មានភាពសាំញ៉ាំ ហេតុនេះដំណើរការរចនាប្លង់ អាចត្រូវបានគេកែសម្រួលបន្ថែមដូចបង្ហាញក្នុងរូបភាព១.២។



រូបភាព១.២ ៖ ដំណើរការរចនាប្លង់សម័យថ្មី

-ដឹងតម្រូវការពីទីផ្សារ

ជាញឹកញយ គេចាប់ផ្តើមរចនាប្លង់នៅពេលគេដឹងថាមានតម្រូវការ ឬមានសក្តានុពលទីផ្សារ សម្រាប់ផលិតផល ឬឧបករណ៍។ វាក៏អាចចេញមកពីតម្រូវការរបស់ក្រុមហ៊ុនក្នុងការធ្វើឲ្យផលិតផលដែលមានស្រាប់កាន់តែប្រសើរឡើង (មានប្រសិទ្ធភាពជាងមុន កាត់បន្ថយតម្លៃផលិត)។

-ការឲ្យនិយមន័យនៃបញ្ហា ឬ តម្រូវការ

វាគឺជាការកំណត់លក្ខណៈបច្ចេកទេសនៃផលិតផលដែលត្រូវរចនាប្លង់ ដែលច្រើនតែមានសរសេរ ជាតុចូល ជាតុចេញ ទំហំ ចរិកលក្ខណៈ និងព្រំដែនកំណត់។ លក្ខណៈបច្ចេកទេសគឺជាតម្រូវការដែលបានកត់ត្រាទុក ដើម្បីគេព្យាយាមបំពេញតម្រូវការនេះដោយការរចនាប្លង់។ លក្ខណៈបច្ចេកទេសច្រើនតែរួមបញ្ចូលរិមាត្រ ប៉ារ៉ាម៉ែត្រនៅពេលម៉ាស៊ីនឬឧបករណ៍ដំណើរការ ឧទាហរណ៍ អានុភាព ល្បឿន សមត្ថភាពការងារ ជាដើម។

-ការសំយោគ

វាគឺជាដំណើរការរួមបញ្ចូលគំនិតជាច្រើនទៅជាទម្រង់ដែលអាចដោះស្រាយបញ្ហានៃតម្រូវការក្នុងការរចនាប្លង់បានកំណត់នៅក្នុងជំហ៊ានមុននេះ។

-ការវិភាគ

ដំណាក់កាលនេះមានប្រើប្រាស់វិទ្យាសាស្ត្រវិស្វកម្ម ដែលជាមុខវិជ្ជារៀននៅក្នុងជំនាញវិស្វកម្ម ដូចជា ស្តាទិក ឌីណាមិក មេកានិកនៃសម្ភារៈបរិក្ខារ លំហូរអង្គធាតុរាវ និងការផ្ទេរកម្តៅ ជាដើម។ ចំណេះដឹងទាំងនេះជួយឲ្យយើងអាចប៉ាន់ស្មានជាមុន ថាតើសម្ភារៈដែលជ្រើសរើសយកទៅប្រើប្រាស់នៅផ្នែកផ្សេងៗរបស់ផលិតផលមានភាពរឹងមាំគ្រប់គ្រាន់ដែរឬទេ? តើវានឹងនៅតែដំណើរការធម្មតានៅសីតុណ្ហភាពដែលផលិតផលនេះត្រូវប្រើប្រាស់ដែរឬទេ? ការសំយោគនិងការវិភាគ អាចត្រូវធ្វើសារចុះឡើង ហើយជាធម្មតាត្រូវធ្វើការសំយោគមុន ទើបវិភាគក្រោយ។

-ការកែសម្រួលឲ្យកាន់តែល្អ (Optimization)

ជាដំណើរការធ្វើសារចុះឡើងដើម្បីឲ្យផលិតផលកាន់តែល្អ តាមរយៈការថ្លឹងថ្លែងលក្ខណៈវិនិច្ឆ័យជាច្រើនដែលជាញឹកញយផ្ទុយគ្នាដើម្បីទទួលបានផលិតផលចុងក្រោយដែលល្អ ឧទាហរណ៍ ដើម្បីឲ្យផលិតផលល្អត្រូវប្រើប្រាស់សម្ភារៈល្អ ឬ បច្ចេកវិទ្យាថ្មី ដែលធ្វើឲ្យតម្លៃផលិតផលខ្ពស់ ហួសលទ្ធភាពអ្នកប្រើប្រាស់ភាគច្រើនក្នុងការទិញយកមកប្រើប្រាស់ ហេតុនេះគេត្រូវរកចំណុចសមស្របណាមួយដែលល្អទាំងកត្តាតម្លៃនិងគុណភាព។

-ការវាយតម្លៃ

គឺជាដំណើរការដើម្បីផ្ទៀងផ្ទាត់ថាផលិតផលដែលបានរចនាប្លង់ហើយថ្មី បំពេញបានតម្រូវការដែលចង់បានហើយឬនៅ? គេច្រើនធ្វើតេស្តផលិតផលដំបូង (prototype) និងធ្វើការស្រាវជ្រាវទីផ្សារ។ ការធ្វើ តេស្តនេះត្រូវអនុវត្តន៍តាមបែបបទនៃការធ្វើតេស្តណាមួយដែលគេទទួលស្គាល់ ដើម្បីលទ្ធផលអាចយកជាការបាន និងមិនប្រែប្រួលនៅពេលធ្វើតេស្តឡើងវិញ។

ការរចនាប្លង់ផលិតផលដែលមានលក្ខណៈធំនិងស្មុគស្មាញ ច្រើនធ្វើឡើងដោយក្រុមរចនាប្លង់ ដែលមានសមាសភាពចូលរួមពីអ្នកពាក់ព័ន្ធ ដូចជា ផ្នែកលក់ ផ្នែករកទីផ្សារ ផ្នែកផលិត ជាដើម ដើម្បីការគិតបានគ្រប់ជ្រុងជ្រោយ និងធានាថាផលិតផលដែលរចនាប្លង់ចេញមកត្រូវនឹងតម្រូវការរបស់អ្នកប្រើប្រាស់ ងាយស្រួលក្នុងការផលិត ប្រើប្រាស់និងថែទាំ និងមានតម្លៃសមស្រប។

១.១.៣ ប្រភេទផ្សេងៗនៃដំណើរការរចនាប្លង់

ការរចនាប្លង់អាចចែកចេញជាបីប្រភេទដូចខាងក្រោម ៖

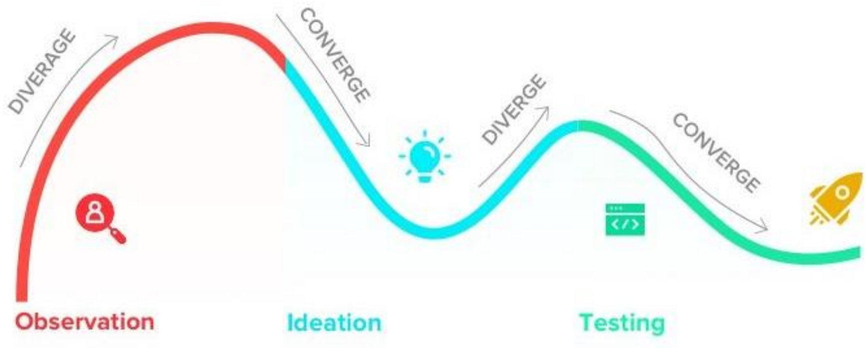
-ការរចនាប្លង់ដើម ឬការរចនាប្លង់ថ្មី (Original Design ឬ New Design): ការរចនាប្លង់ដែលថ្មី មិនសូវស្រដៀងនឹងផលិតផលដែលមានស្រាប់ ការរចនាប្លង់ដែលទទួលបានកម្មសិទ្ធិបញ្ញា។

-ការរចនាប្លង់កែសម្រួល (Transitional Design ឬ Adaptive Design): កែលម្អផលិតផលមានស្រាប់ឲ្យកាន់តែប្រសើរ

-ការរចនាប្លង់ពង្រីក (Extensional Design ឬ Developmental Design): បង្កើនទំហំផលិតផលដែលប្រើប្រាស់ល្អស្រាប់ហើយ ឲ្យធំជាងមុន

១.១.៤ Human-centered Design

Human-centered Design គឺជាមធ្យោបាយក្នុងការដោះស្រាយបញ្ហា គេច្រើនប្រើប្រាស់នៅក្នុងការរចនាប្លង់ ឬ ក្របខណ្ឌគ្រប់គ្រង ដែលផ្តល់ដំណោះស្រាយបញ្ហាដោយរួមបញ្ចូលទិដ្ឋភាពដែលពាក់ព័ន្ធនឹងមនុស្ស (អ្នកប្រើប្រាស់) នៅក្នុងដំណើរការដោះស្រាយបញ្ហា សូមមើលរូបភាព ១.៣។ និយាយឲ្យខ្លីគឺជាការរចនាប្លង់ដែលគិតដល់តម្រូវការរបស់អ្នកប្រើប្រាស់ ដើម្បីឲ្យងាយស្រួលប្រើប្រាស់។ ដំណើរការរចនាប្លង់នេះមានបីវគ្គ ហើយចំណុចមួយចំនួនស្រដៀងនឹង ដំណើរការរចនាប្លង់ដែលបានបង្ហាញកន្លងមកដែរ។



រូបភាព ១.៣៖ Human-centered Design

ប្រភព ៖ https://www.netsolutions.com/insights/wp-content/uploads/2019/08/Three-Stages-of-Human-Centered-Design.jpg

-សង្កេត

អ្នករចនាប្លង់ត្រូវឆ្លើយសំនួរទាំងនេះ តើនរណានឹងប្រើប្រាស់ផលិតផលនេះ? តើចង់ជួយអ្នក ប្រើប្រាស់ឲ្យបានសម្រេចគោលដៅអ្វី? ជាធម្មតា យើងអាចឆ្លើយសំនួរនេះបានតាមរយៈការធ្វើការស្រាវជ្រាវទីផ្សារផលិតផលដែលមានស្រាប់។ តើផលិតផលនៅលើទីផ្សារមានចំណុចខ្វះខាតអ្វីខ្លះ? ចំណុចអ្វីខ្លះដែលអ្នកប្រើប្រាស់គិតថាមិនល្អ គួរកែសម្រួល?

-បង្កើតគំនិត

វគ្គនេះមានការជួបប្រជុំក្រុមរចនាប្លង់ដើម្បីធ្វើព្យុះគំនិត (brainstorming) ដើម្បីបង្កើតផលិតផលថ្មី អាចដោះស្រាយបញ្ហាដែលបានកំណត់នៅក្នុងវគ្គមុន។ គំនិតអាចមានច្រើន ប៉ុន្តែត្រូវជ្រើសរើសគំនិតល្អមួយក្នុងចំណោមគំនិតជាច្រើន។

-ការធ្វើតេស្ត

ជាផ្នែកសំខាន់នៃដំណើរការ Human-centered Design។ គេត្រូវធ្វើតេស្តដើម្បីឲ្យប្រាកដថាផលិតផលមានប្រយោជន៍សម្រាប់អ្នកប្រើប្រាស់។ សំនួរអាចមាន តើអ្នកប្រើប្រាស់ចេះប្រើផលិតផលនេះដែរឬទេ? តើអ្នកប្រើប្រាស់ពេញចិត្តដែរឬទេ? តើវាបានដោះស្រាយបញ្ហាដែលចង់ឲ្យវាដោះស្រាយដែរឬទេ? ក្រោយពេលធ្វើតេស្ត ត្រូវវាយតម្លៃចំណុចខ្វះខាត កែលម្អ ហើយធ្វើដំណើរការនេះសារឡើងវិញ រហូតដល់ទទួលបានអ្វីដែលយើងចង់បាន។

១.២ ឧបករណ៍សម្រាប់ប្រមូលតម្រូវការរបស់អ្នកប្រើប្រាស់

ការកំណត់តម្រូវការរបស់អ្នកប្រើប្រាស់ជាដំណាក់កាលដ៏ចាំបាច់ក្នុងការរចនាប្លង់។ គេអាចប្រើប្រាស់ឧបករណ៍ ដូចជា ព្យុះគំនិត (brainstorming) ដើម្បីជួយក្នុងការប្រមូលតម្រូវការនេះ។

១.២.១ ព្យុះគំនិត (brainstorming)

Brainstroming ជាការចូលរួមប្រជុំជាក្រុមពីអ្នកដែលមានជំនាញផ្សេងៗគ្នា គោលបំណងដោះស្រាយបញ្ហាអ្វីមួយ។ គោលបំណងគឺដើម្បីប្រមូលគំនិតឲ្យបានច្រើន មិនគិតល្អ ឬ មិនល្អទេ ហេតុនេះមានច្បាប់មួយដែលហាមមិនឲ្យរិះគន់គំនិតអ្នកដទៃដែលចូលរួមក្នុងការបញ្ចេញគំនិត។ គេត្រូវមានអ្នកកត់ត្រាម្នាក់ ដើម្បីកត់អ្វីដែលសមាជិកក្រុមបានលើកឡើង។ គំនិតដែលប្រមូលបានត្រូវយកទៅវិភាគបន្ថែមទៀត ដូចជា ដាក់គំនិតស្រដៀងគ្នាជាក្រុមជាដើម និងកំណត់ចំណុចសំខាន់ៗដែលក្រុមបានលើកឡើង។

១.២.១ វិធីជំរុញឲ្យអ្នកចូលរួមបញ្ចេញគំនិត

នៅក្នុងដំណើរការព្យុះគំនិត សមាជិកខ្លះមិនហ៊ាន ឬ មិនសូវបញ្ចេញយោបល់ ប៉ុន្តែក៏អាចមានអ្នកចូលរួមខ្លះដែលនិយាយច្រើនពេក ដែលធ្វើឲ្យអ្នកផ្សេងមិនមានឱកាសបញ្ចេញយោបល់។ ដើម្បីដោះស្រាយបញ្ហានេះ ត្រូវពន្យល់ពីហេតុផលនិងច្បាប់នៃ brainstorming មុននឹងចាប់ផ្តើមធ្វើ ។ ច្បាប់អាចបង្កើតដោយអ្នកចូលរួម ដោយជួបពិភាក្សាគ្នាមុន និងព្រមព្រៀងគ្នា តើអ្វីដែលមិនត្រូវធ្វើនៅពេលធ្វើព្យុះគំនិត។

១.២.២ ច្បាប់សំខាន់ៗដែលត្រូវគោរពតាម

ដើម្បីឲ្យការធ្វើព្យុះគំនិតដំណើរការល្អ ការធ្វើគួរគិតចំណុចខាងក្រោមនេះ ៖

- សមាជិកក្រុមត្រូវមានជំនាញផ្សេងៗគ្នា ដើម្បីឲ្យមានគំនិតច្រើន និងគិតគូរបានគ្រប់ជ្រុងជ្រោយ
- មិនគួរមានសមាជិកច្រើនជាង១០នាក់ សមាជិកលើសពីនេះ អាចពិបាកក្នុងការគ្រប់គ្រង
- មិនអនុញ្ញាតឲ្យរិះគន់គំនិតអ្នកដទៃ ការរិះគន់និងធ្វើឲ្យអ្នកបញ្ចេញយោបល់ឈប់បញ្ចេញគំនិត ដែលគំនិតនោះអាចជាគំនិតល្អ
- ត្រូវរៀបចំ flip chart, post-it ក្រដាសតូចៗ ហ្វឺតសម្រាប់សរសេរ។ល។ ដើម្បីឲ្យអ្នកចូលរួមអាចសរសេរគំនិតរបស់គេ
- បញ្ជាក់ប្រធានបទឲ្យបានច្បាស់លាស់ ប្រសិនបើមិនបានបញ្ជាក់ប្រធានបទច្បាស់លាស់ទេ ការពិភាក្សាអាចងាកចេញពីប្រធានបទ
- កំណត់ពេលវេលា (មិនគួរយូរជាង ៦០ ទៅ ៩០ នាទី) យូរជាងនេះសមាជិកនឹងបាត់បង់ការផ្តោតអារម្មណ៍

- ជ្រើសរើសអ្នកបង្ហាញលទ្ធផល ត្រូវជ្រើសរើសតំណាងម្នាក់ដើម្បីធ្វើបទបង្ហាញអ្វីដែលបានពិភាក្សា សមាជិកផ្សេងទៀតអាចពិនិត្យមើល និងកែលម្អប្រសិនបើមានចំណុចខ្វះខាត

១.២.៣ ប្រភេទឧបករណ៍ប្រើប្រាស់

១.២.៣.១ ព្យុះគំនិតប្រើប្រាស់ផ្ទាំងក្រដាសធំ (Flip Chart Brainstorming)

រូបភាព ១.៤ ៖ ព្យុះគំនិតប្រើប្រាស់ផ្ទាំងក្រដាសធំៗ

ព្យុះគំនិតបែបនេះ អ្នកកត់ត្រាការពិភាក្សាប្រើប្រាស់ផ្ទាំងក្រដាសធំៗ (A0) ដើម្បីកត់ត្រាគំនិតសំខាន់ៗដែលអ្នកចូលរួមបានលើកឡើង ដើម្បីទុកយកមកធ្វើបទបង្ហាញ និងពិភាក្សាបន្ថែមទៀតនៅពេលក្រោយ។ ការកត់ត្រាគំនិតនៅលើក្រដាស អាចជួយឲ្យអ្នកចូលរួមងាយស្រួលក្នុងការពិនិត្យមើលឡើងវិញ គំនិតដែលអ្នកចូលរួមបានលើកឡើង និងអាចនាំឲ្យនឹកឃើញគំនិតផ្សេងទៀត។ គំនិតដែលលើកឡើង គេអាចចាត់វាជាក្រុមបន្តទៀត ដោយដាក់គំនិតស្រដៀងគ្នាក្នុងក្រុមតែមួយ ឬអាចរកទំនាក់ទំនងពីក្រុមមួយទៅក្រុមមួយទៀត។

១.២.៣.២ ព្យុះគំនិតប្រើប្រាស់សន្លឹកក្រដាសតូចៗ (Post-it Brainstorming)

រូបភាព ១.៥ ៖ ព្យុះគំនិតដោយប្រើប្រាស់ Post-it

ក្នុងពេលធ្វើព្យុះគំនិតនេះ អ្នកសម្របសម្រួលចែកសន្លឹកក្រដាសតូចៗ (អាចមានពណ៌ផ្សេងគ្នា និងមានការស្អិតដើម្បីបិទវាទៅលើផ្ទាំងក្រដាសធំ ឬលើក្ដារខៀន) ដើម្បីឲ្យអ្នកចូលរួមអាចសរសេរគំនិត ឬយោបល់របស់ខ្លួននៅលើនោះតាមប្រធានបទដែលអ្នកសម្របសម្រួលលើកឡើងសម្រាប់ការពិភាក្សា។ ជាធម្មតា គំនិតមួយសរសេរលើក្រដាសមួយសន្លឹក។ ចំណុចល្អនៃវិធីសាស្ត្រនេះ អ្នកចូលរួមទាំងអស់អាចមានភាពក្លាហានក្នុងការសរសេរគំនិតរបស់ខ្លួន ដោយមិនសូវបារម្ភថាជាគំនិតល្អ ឬមិនសូវល្អ និងមួយទៀត ផ្ទាំងក្រដាសតូចៗនេះ ងាយស្រួលលើកពីកន្លែងមួយទៅដាក់កន្លែងមួយទៀតនៅពេលចាត់វាជាក្រុម។

១.២.៥ វិធីជួយឲ្យអ្នកចូលរួមមានគំនិត

នៅក្នុងការពិភាក្សាជាក្រុម នៅពេលអ្នកចូលរួមមិនសូវមានគំនិតក្នុងការពិភាក្សា អ្នកសម្របសម្រួលអាចប្រើប្រាស់វិធីសាស្ត្រខាងក្រោមនេះដើម្បីបំផុសគំនិតរបស់អ្នកចូលរួម។

- Mind mapping (រកទំនាក់ទំនងនៃគំនិត) រកទំនាក់ទំនងនៃគំនិតដែលអ្នកចូលរួមពិភាក្សាលើកឡើង
- Right braining (ប្រើប្រាស់ខួរក្បាលផ្នែកខាងស្ដាំ)
- Provocative action (ត្រលប់កៅអីផ្ងាជើង ដើម្បីឲ្យអ្នកចូលរួមគិតប្លែកៗ)
- Break and build (បំបែក រួចផ្គុំបញ្ចូលគ្នាវិញ)
- អ្នកសុទិដ្ឋិនិយម និង អ្នកទុទិដ្ឋិនិយម ចែកជាពីរក្រុម មួយក្រុមលើកគំនិតសុទិដ្ឋិនិយម និងមួយក្រុមទៀតលើកគំនិតទុទិដ្ឋិនិយម
- ភាពចៃដន្យ

១.៣ ការធ្វើតេស្តម៉ាស៊ីននិងឧបករណ៍កសិកម្ម

នៅក្នុងដំណើរការរចនាប្លង់ បន្ទាប់ពីផលិតផលិតផលដំបូង (Prototype) រួចហើយ គេត្រូវយកផលិតផលដំបូងនេះទៅធ្វើតេស្តដើម្បីពិនិត្យមើលថាវាដំណើរការដូចអ្វីដែលអ្នករចនាប្លង់ចង់ឲ្យវាអាចធ្វើបានដែរឬទេ? គឺជាការយកវាទៅប្រើប្រាស់សាកល្បង និងវាស់វែងប៉ារ៉ាម៉ែត្រសំខាន់ៗដែលបានកំណត់នៅក្នុងលក្ខណៈបច្ចេកទេសនៅក្នុងដំណាក់ខាងដើម នៃដំណើរការរចនាប្លង់។

១.៣.១ គោលបំណងនៃការធ្វើតេស្ត

ការធ្វើតេស្តមានគោលបំណងប្រៀបធៀប ឬ ផ្ទៀងផ្ទាត់ប្រសិទ្ធភាពការងាររបស់ម៉ាស៊ីន ឬ ឧបករណ៍ធៀបទៅនឹងលក្ខណៈបច្ចេកទេស ដើម្បីដឹងថាម៉ាស៊ីន ឬ ឧបករណ៍អាចបំពេញតម្រូវការរបស់អ្នកប្រើប្រាស់ ឬ ជាប់ស្តង់ដារដែរឬទេ?

ការធ្វើតេស្ត (testing) និងការវាយតម្លៃ (evaluation) មានភាពខុសគ្នាមួយចំនួន។ ការធ្វើតេស្តនិងវាយតម្លៃមានភាពខុសគ្នាត្រង់ថា ការធ្វើតេស្តគឺធ្វើនៅក្នុងមន្ទីរពិសោធន៍ដែលងាយស្រួលក្នុងការកំណត់កត្តាផ្សេងៗដែលមិនពាក់ព័ន្ធឲ្យនៅថេរ ហើយពិនិត្យមើលតែលើប៉ារ៉ាម៉ែត្រដែលគេចង់ដឹងតែប៉ុណ្ណោះ។ ដោយសារមូលហេតុនេះ លទ្ធផលនៅការធ្វើពីម្តងទៅម្តងទៀត ឬពីមន្ទីរពិសោធន៍មួយទៅមួយទៀត អាចទទួលបានលទ្ធផលដូចគ្នា ករណីធ្វើទៅលើម៉ាស៊ីនឬឧបករណ៍តែមួយ។ ចំណែកឯការវាយតម្លៃវិញគឺធ្វើនៅលើទីវាលដែលជាកន្លែងម៉ាស៊ីន ឬ ឧបករណ៍ ត្រូវប្រើប្រាស់ជាក់ស្តែង។ លទ្ធផលវាយតម្លៃប្រែប្រួលពីកន្លែងមួយទៅមួយទៀត ដោយសារមានកត្តាផ្សេងៗជាច្រើនដែលពិបាកទទួលបានដូចគ្នា។

១.៣.២ Test Code

វាគឺជាវិធីសាស្ត្រធ្វើតេស្តដែលគេបានបង្កើតឡើងនិងចងក្រងជាឯកសារ ដើម្បីធានាថាការធ្វើតេស្តត្រូវបានអនុវត្តន៍ដូចគ្នា ដើម្បីធានាថាលទ្ធផលទទួលបានដូចគ្នានៅពេលធ្វើតេស្តលើម៉ាស៊ីនដំបូង (prototype) ប្រភេទដូចគ្នា ហើយលទ្ធផលធ្វើតេស្តនៅកន្លែងផ្សេងគ្នា អាចប្រៀបធៀបគ្នាបាន។ ការធ្វើតេស្តត្រូវអនុវត្តតាម test code ដូចជា RNAM ឬ សៀវភៅរបស់ FAO នៅក្នុងឯកសារយោងមេរៀនរបស់យើងជាដើម។

- https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjBx__pt-LpAhVZdCsKHfjhCa0QFjAGegQICBAB&url=http%3A%2F%2Fwww.fao.org%2F3%2Fa-t1841e.pdf&usg=AOvVaw2-PubE4FfYsl2TbmE3Rrqy

១.៣.៣ ប៉ារ៉ាម៉ែត្រដែលត្រូវវាស់វែង

ប៉ារ៉ាម៉ែត្រដែលត្រូវវាស់វែងនៅពេលធ្វើតេស្ត ឬ វាយតម្លៃ គឺអាស្រ័យនឹងលក្ខណៈបច្ចេកទេសរបស់ម៉ាស៊ីន ឬ ឧបករណ៍ ឧទាហរណ៍ ទម្ងន់ វិមាត្រ (បណ្តោយ ទទឹង កម្ពស់) អានុភាព ល្បឿន។ល។

ឧបករណ៍វាស់វែងត្រូវធានាថាវាស់បានត្រឹមត្រូវ ហេតុនេះមុនប្រើប្រាស់ គេត្រូវធ្វើ calibration ជាមុនសិន ដោយប្រៀបធៀបលទ្ធផលវាស់វែងដោយឧបករណ៍នេះ ទៅនឹងឧបករណ៍ដែលគេជឿជាក់ថាវាវាស់វែងបានត្រឹមត្រូវ។ ឧទាហរណ៍ ទែរម៉ូម៉ែត្រ ប្រសិនបើវាស់វែងទឹកកកកំពុងរលាយវាបង្ហាញសីតុណ្ហភាព 0 អង្សាសេ ហើយវាស់វែងទឹកកំពុងពុះវាបង្ហាញសីតុណ្ហភាព១០០ អង្សាសេ អាចបញ្ជាក់ថាការវាស់វែងរបស់វាត្រឹមត្រូវ។ ដើម្បីធ្វើ calibration ឧបករណ៍វាស់វែង គេត្រូវបញ្ជូនវាទៅកាន់មន្ទីរពិសោធន៍មាត្រាសាស្ត្រដែលមានការទទួលស្គាល់។ ការវាស់វែងដែលត្រឹមត្រូវមានភាពចាំបាច់ណាស់ ក្នុងការស្រាវជ្រាវវិទ្យាសាស្ត្រ និងសុវត្ថិភាព។ ឧទាហរណ៍ ឧបករណ៍វាស់សីតុណ្ហភាពរបស់ម៉ាស៊ីនផលិតចំហាយទឹកសម្រាប់ប្រើប្រាស់ក្នុងការកែច្នៃម្ហូបអាហារ ឬ ឧស្សាហកម្មផ្សេងៗ ប្រសិនបើការវាស់វែងរបស់វាខុស អាចធ្វើឲ្យប៉ះពាល់ដល់គុណភាពម្ហូបអាហារ ឬ អាចផ្ទុះ ដោយសារសីតុណ្ហភាពឡើងខ្ពស់ពេក ប៉ុន្តែប្រតិបត្តិករមិនបានដឹង ដោយសារឧបករណ៍វាស់សីតុណ្ហភាពបង្ហាញលទ្ធផលខុស។

១.៣.៤ អ្នកប្រើប្រាស់របាយការណ៍ធ្វើតេស្ត

ក្រោយពេលបញ្ចប់ការធ្វើតេស្ត អ្នកធ្វើតេស្តត្រូវធ្វើរបាយការណ៍ ដែលអាចប្រើប្រាស់ដោយ អ្នករចនាប្លង់ ឬ អ្នកអភិវឌ្ឍម៉ាស៊ីន ឬ ឧបករណ៍ ឬ អ្នកផ្តល់វិញ្ញាបនបត្រ ដើម្បីបញ្ជាក់ថាម៉ាស៊ីន ឬ ឧបករណ៍ ដែលបានធ្វើតេស្តហើយ ជាប់ស្តង់ដា ណាមួយ។ ឧទាហរណ៍ នៅកន្លែងឆៀករថយន្ត នៅពេលឆ្លងកាត់ការធ្វើតេស្តលក្ខណៈបច្ចេកទេសរបស់រថយន្តរួចហើយ រថយន្តដែលជាប់ស្តង់ដា គេនឹងចេញស្ទិកគ័រឆៀកសម្រាប់មកបិទនៅលើរថយន្តដើម្បីបញ្ជាក់ថា វាមានលក្ខណៈគ្រប់គ្រាន់អាចចរាចរនៅលើផ្លូវថ្នល់បាន។

ជំពូកទី២
ប្រព័ន្ធវាស់វែង កត់ត្រាទិន្នន័យ និងបញ្ជាស្វ័យប្រវត្តិ

២.១ ធាតុនៃគ្រឿងអេឡិកត្រូនិក

នៅក្នុងឧបករណ៍វាស់វែង និងកត់ត្រាទិន្នន័យ ដែលប្រើប្រាស់ក្នុងការធ្វើតេស្ត និងតាមដានដំណើរការក្នុងរោងចក្រឧស្សាហកម្ម ឬ រោងចក្រកែច្នៃក្នុងវិស័យកសិកម្ម មានគ្រឿងអេឡិចត្រូនិកផ្សំដោយធាតុតូចៗដូចខាងក្រោម ៖

- រេស៊ីស្តង់ (resistors)
- ឌីយ៉ូត (diodes)
- កុងដង់សាទ័រ (capacitors)
- ត្រង់ស៊ីស្ទ័រ (transistors)
- គ្រឿងអេឡិចត្រូនិកប្រើពន្លឺ (opto-electronic components)។ល។

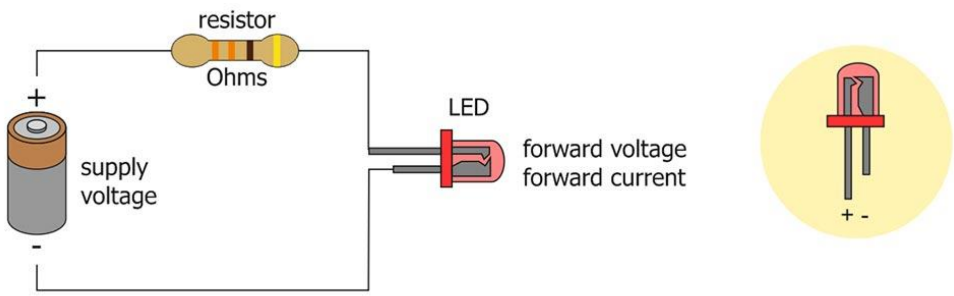
២.១.១ រ៉េស៊ីស្តង់

រ៉េស៊ីស្តង់ជាធាតុនៃគ្រឿងអេឡិកត្រូនិកមានតួនាទីទប់លំហូរអេឡិកត្រុង កាត់បន្ថយតង់ស្យុង ឬ អាំងតង់ស៊ីតេចរន្ត ដោយបំលែងថាមពលអគ្គីសនីទៅជាកម្តៅ។ ជាធម្មតា ខ្នាតរបស់រ៉េស៊ីស្តង់គិតជា អូម (Ω) គីឡូអូម (kΩ) ឬ ម៉េកូម (MΩ)។ រ៉េស៊ីស្តង់មានច្រើនប្រភេទ មានរ៉េស៊ីស្តង់ថេរ ដែលប្រើប្រាស់ពណ៌សម្រាប់ជាកូដនៃរ៉េស៊ីស្តង់ ដូចបង្ហាញក្នុងរូបភាព ២.១ និងរ៉េស៊ីស្តង់ប្រែប្រួល ដូចជា potentiometer ជាដើម នៅក្នុងរូបភាព ២.២។

រូបភាព ២.១ ៖ ការប្រើប្រាស់ឆ្នូតពណ៌ដើម្បីបង្ហាញតម្លៃរ៉េស៊ីស្តង់
ប្រភព៖ https://www.explainthatstuff.com

រូបភាព ២.២ ៖ រ៉េស៊ីស្តង់ប្រែប្រួល

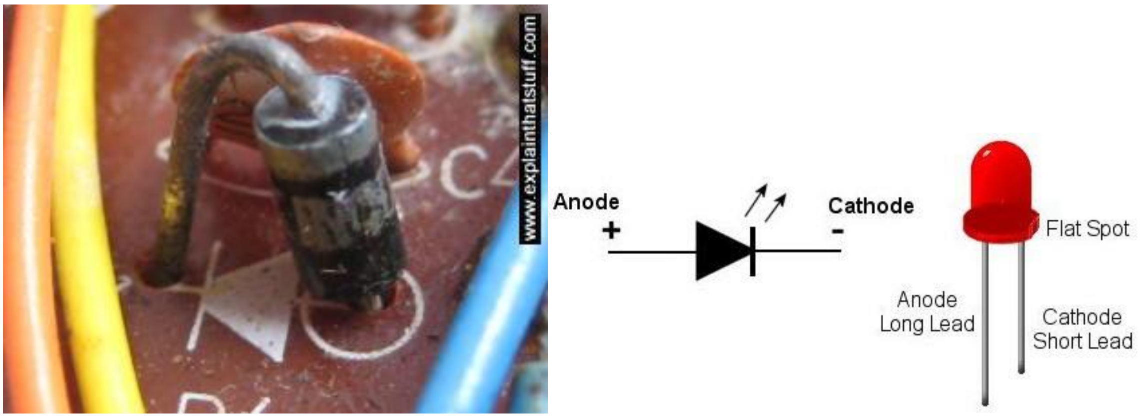
រ៉េស៊ីស្តង់នៅក្នុងរូបភាព ២.៣ គេប្រើប្រាស់វាដើម្បីបញ្ចុះតង់ស្យុងរបស់ថ្មពិល ទៅដល់កម្រិតតង់ស្យុងដែលអាចផ្តល់ថាមពលទៅ LED ដើម្បីបញ្ចេញពន្លឺ។ ប្រសិនបើ តង់ស្យុងខ្ពស់ពេក LED នឹងឆេះខូច ដើម្បីដោះស្រាយបញ្ហានេះ គេត្រូវជ្រើសរើសរ៉េស៊ីស្តង់សមស្រប ដើម្បីបញ្ចុះតង់ស្រុងរបស់ថ្មពិលឲ្យនៅល្មមតែធ្វើឲ្យ LED បញ្ចេញពន្លឺ។ ប្រសិនបើតង់ស្យុងរបស់ថ្មពិលខ្ពស់ជាងតង់ស្យុងដែល LED ត្រូវការដើម្បីដំណើរ គេត្រូវត LED ជាស៊េរីជាមួយរេស៊ីស្តង់ ដើម្បីបន្ថយតង់ស្យុងនៅជើងទាំងពីររបស់ LED។ នៅពេលយើងតភ្ជាប់រ៉េស៊ីស្តង់ជាស៊េរីជាមួយ LED បើជ្រើសរើសរ៉េស៊ីស្តង់ធំខ្លាំងពេក ធ្វើឲ្យមានទម្លាក់តង់ស្យុងខ្លាំង បណ្តាលឲ្យ LED ភ្លឺតិច ឬ មិនភ្លឺតែម្តង។



រូបភាព ២.៣ ៖ ការប្រើប្រាស់រ៉េស៊ីស្តង់ដើម្បីបញ្ចុះតង់ស្យុងរបស់ថ្មពិល

២.១.២ ឌីយ៉ូត

ឌីយ៉ូតជាប្រភេទអង្គធាតុពាក់កណ្តាលចម្លង (semiconductor) វាស្រដៀងនឹងផ្លូវឯកទិសដែរ អាចឲ្យអេឡិចត្រុងរត់ទិសដៅតែមួយប៉ុណ្ណោះ (មិនអាចរត់បញ្ច្រាសបានទេ)។ វាមានប៉ូលវិជ្ជមាន (អាណូត) និង ប៉ូលអវិជ្ជមាន (កាតូត) ហើយចម្លងចរន្តតែនៅពេលប៉ូលរបស់វាតភ្ជាប់ត្រឹមត្រូវតែប៉ុណ្ណោះ។ គេប្រើប្រាស់វាសម្រាប់ប្តូរចរន្តឆ្លាស់ទៅជាចរន្តជាប់។ ឌីយ៉ូតបញ្ចេញពន្លឺ (Light-emitting Diode-LED) ប្រើប្រាស់ជាភ្លើងសញ្ញានៅក្នុងឧបករណ៍អេឡិកត្រូនិក ដោយសារវាស៊ីថាមពលអគ្គីសនីតិចជៀបទៅនឹងអំពូលភ្លើង។ ចំណែកឯ ឌីយ៉ូតពន្លឺ (Photo Diode) បង្កើតជាចរន្តអគ្គីសនីនៅពេលគេបញ្ចាំងពន្លឺទៅលើវា។



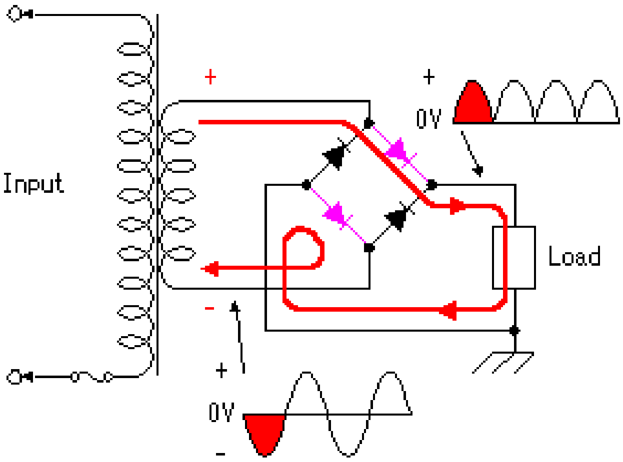
រូបភាព ២.៤ ៖ ឌីយ៉ូតធម្មតានិងឌីយ៉ូតបញ្ចេញពន្លឺ

ប្រភព ៖ https://www.explainthatstuff.com

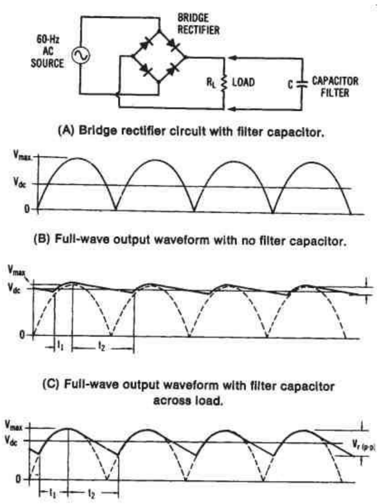
២.១.៣ កុងដង់សាទ័រ

កុងដង់សាទ័រ (capacitor) ប្រើប្រាស់សម្រាប់ស្តុកទុកនិងបញ្ចេញចរន្តអគ្គីសនី។ ខ្នាតរបស់វាច្រើនគិតជា មីក្រូហ្វារ៉ាត (µF) ណាណូហ្វារ៉ាត (nF) និងពីកូហ្វារ៉ាត (pF)។ ប្រភេទរបស់វាមានប្រភេទមានប៉ូលវិជ្ជមាននិងអវិជ្ជមាន និងប្រភេទគ្មានប៉ូលដូចបង្ហាញក្នុងរូបភាព ២.៥។ គេក៏ប្រើប្រាស់វាជាឧបករណ៍សម្រាប់កំណត់ពេលវេលា ប្រើក្នុងចម្រោះ (filter) សម្រាប់ដកសំលេងរំខាន (noise) ចេញ ដូចបង្ហាញនៅក្នុងរូបភាព ២.៦។ នៅក្នុងជុំភ្លើង ឬឧបករណ៍សាកថ្មទូរស័ព្ទច្រើនប្រើប្រាស់ឌីយ៉ូតជា bridge rectifier ដើម្បីបម្លែងចរន្តឆ្លាស់ឲ្យរត់ទិសដៅតែមួយ ដោយប្រើប្រាស់ឌីយ៉ូតចំនួន៤។ ចរន្តឆ្លាស់ដែលបានប្តូរទិសដៅរួចហើយ នៅមានទម្រង់ជារលកនៅឡើយ នៅពេលតភ្ជាប់កុងដង់សាទ័រជាខ្នែងជាមួយបន្ទុក វាជួយឲ្យរលកចរន្តអគ្គីសនីមិនសូវមានភាពខុសគ្នាខ្លាំងដូចក្នុងរូបភាព២.៧ ដោយសារកុងដង់សាទ័រផ្ទុកចរន្តអគ្គីសនីនៅពេលតង់ស្យុងខ្លាំង និងបញ្ចេញចរន្តអគ្គីសនីមកវិញនៅពេលតង់ស្យុងអគ្គីសនីបញ្ចេញពី bridge rectifier ខ្សោយ។

រូបភាព ២.៥ ៖ ប្រភេទផ្សេងៗនៃកុងដង់សាទ័រ



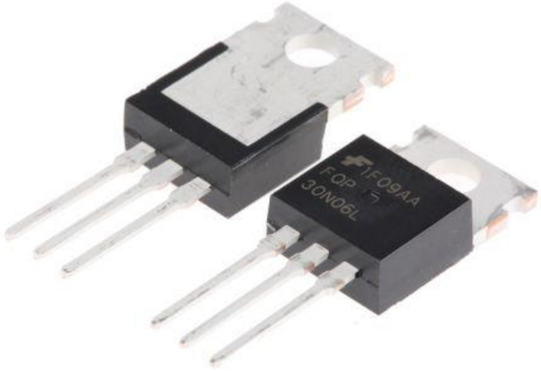
រូបភាព ២.៦ ៖ ដំណើរការបម្លែងពីចរន្តឆ្លាស់ទៅជាចរន្តជាប់ដោយប្រើប្រាស់ឌីយ៉ូត



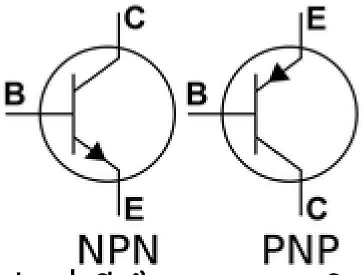
រូបភាព ២.៧ ៖ ការប្រើប្រាស់កុងដង់សាទ័រដើម្បីកាត់បន្ថយរលកចរន្តអគ្គិសនីនៅក្នុងជុំភ្លើង

២.១.៤ ត្រង់ស៊ីស្ទ័រ

ត្រង់ស៊ីស្ទ័រ (ដូចបង្ហាញក្នុងរូបភាព ២.៨) ជាជាតុដ៏សំខាន់នៃកុំព្យូទ័រ អាចប្រើប្រាស់ជាកុងតាក់សម្រាប់បើកបិទចរន្តអគ្គីសនី ដំឡើងចរន្តអគ្គីសនី (អំភ្លី) ប្រើប្រាស់ជា memory នៅក្នុងកុំព្យូទ័រ។ នៅពេលតត្រង់ស៊ីស្ទ័រច្រើនបញ្ចូលគ្នាវាអាចបង្កើតបានជា logic gates ដែលជាគ្រឿងផ្គុំដ៏សំខាន់នៃឌីជីថលអេឡិកត្រូនិក។ រូបភាព ២.៩ ជានិមិត្តសញ្ញាសម្គាល់ត្រង់ស៊ីស្ទ័រ Bipolar Junction Transistor (BJT) ប្រភេទ NPN និង PNP។ ជើងរបស់វាមានបី Emitter (E), Collector (C) និង Base (B)។



រូបភាព ២.៨ ៖ ត្រង់ស៊ីស្ទ័រ

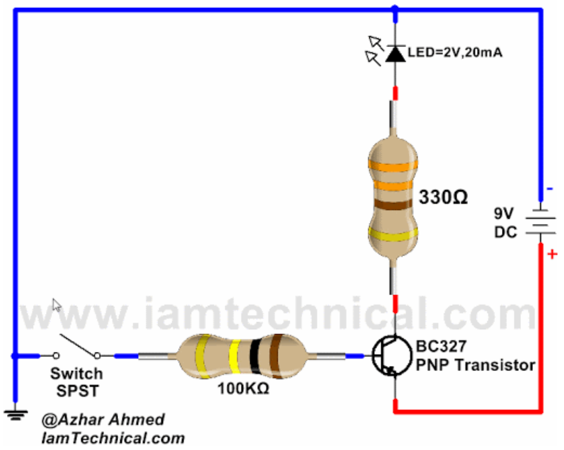


រូបភាព ២.៩ ៖ និមិត្តសញ្ញារបស់ត្រង់ស៊ីស្ទ័រប្រភេទ NPN និង PNP

ប្រភព ៖ https://cdn11.bigcommerce.com/s-fccf0zqr3o/images/stencil/190x250/uploaded_images/shoptransmitter-npn-transistor-pnp-transistor.png?t=1544445403

ក្នុងរូបភាព ២.១០ ជាឧទាហរណ៍នៃការប្រើប្រាស់ត្រង់ស៊ីស្ទ័រជាកុងតាក់។ ត្រង់ស៊ីស្ទ័រដែលប្រើប្រាស់ជាប្រភេទ PNP។ នៅពេលជើង base របស់វាមានតង់ស្យុងទាបជាងតង់ស្យុងថ្មពិលដោយសារតជាសេរីជាមួយរេស៊ីស្តង់ 100 kΩ ចរន្តអគ្គីសនីរត់កាត់ត្រង់ស៊ីស្ទ័រ រេស៊ីស្តង់ 330 Ω ឆ្លងកាត់ LED ទៅកាន់ប៉ូលអវិជ្ជមានរបស់ថ្មពិល ធ្វើឲ្យ LED ភ្លឺ។ ពេលបើកកុងតាក់ ជើង base របស់ត្រង់ស៊ីស្ទ័រលែងមានចរន្តអគ្គិសនី

ត្រង់ស៊ីស្ទ័រលែងចម្លងចរន្តអគ្គិសនីឆ្លងកាត់ LED ធ្វើឲ្យវាឈប់ភ្លឺ។ ខុសពីកុងតាក់ម៉េកានិក ត្រង់ស៊ីស្ទ័រអាចប្រើធ្វើជាកុងតាក់បិទបើកល្បឿនលឿន។



រូបភាព ២.១០៖ ការប្រើប្រាស់ត្រង់ស៊ីស្ទ័រ PNP ជាកុងតាក់បើកបិទ LED

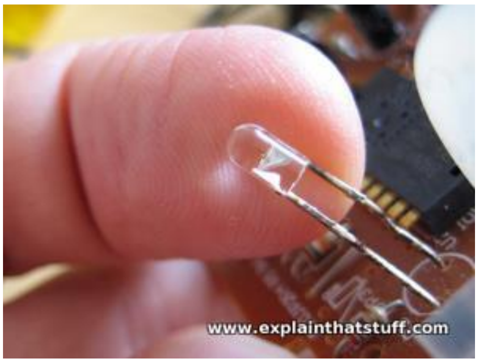
ប្រភព ៖ https://www.iamtechnical.com

២.១.៥ គ្រឿងអេឡិចត្រូនិចប្រើពន្លឺ

ពួកវាអាចបម្លែងពន្លឺទៅជាអគ្គីសនី ឬ អគ្គីសនីទៅជាពន្លឺ មាន ដូចជា ផ្ទាំងសូឡា ឌីយ៉ូតបញ្ចេញពន្លឺ (LED) និងអេក្រង់ (LCD) ជាដើម។ រូបភាព ២.១១ ជាផ្ទាំងសូឡា ដែលប្រើប្រាស់សម្រាប់ផលិតចរន្តអគ្គីសនីពីពន្លឺព្រះអាទិត្យ។ ដំបូងផ្ទាំងសូឡាបំលែងថាមពលពន្លឺព្រះអាទិត្យទៅជាថាមពលអគ្គីសនីជាចរន្តជាប់ ដែលជាមួយប្រព័ន្ធតូចៗមានតង់ស្យុង១២វ៉ុល ឬ ២៤វ៉ុល ដែលអាចប្រើប្រាស់ដោយផ្ទាល់ ឬ បំលែងទៅជាចរន្តឆ្លាស់ដោយប្រើប្រាស់អាំងវែរទ័រ (inverter) មានតង់ស្យុង ២២០វ៉ុល ដែលអាចប្រើប្រាស់ជាមួយឧបករណ៍អគ្គីសនីប្រើប្រាស់ចរន្តឆ្លាស់មានស្រាប់ដូចជា អំពូលភ្លើង កង្ហារ កុំព្យូទ័រ ជាដើម។

រូបភាព ២.១១ ៖ ផ្ទាំងសូឡា

រូបភាព ២.១២ គឺជា LED ដែលប្រើប្រាស់ជាភ្លើងសញ្ញានៅក្នុងគ្រឿងអេឡិកត្រូនិក។ វាអាចមានពណ៌ស ក្រហម បៃតង ឬលឿង។ ដូចឌីយ៉ូតដែរ វាចម្លងអគ្គីសនី និងបញ្ចេញពន្លឺនៅពេលអាណូតរបស់វាតទៅប៉ូលវិជ្ជមាន និងកាតូតតទៅប៉ូលអវិជ្ជមានរបស់ជនិតាដែលមានតង់ស្យុងពី ១.៥ វ៉ុល ទៅ ២វ៉ុល។ នៅពេលតង់ស្យុងកើនឡើងធ្វើឲ្យកម្រិតភ្លឺរបស់ LED កើនឡើងដែរ ប៉ុន្តែបើប្រើប្រាស់តង់ស្យុងខ្ពស់ពេកអាចធ្វើឲ្យ LED ឆេះខូច។



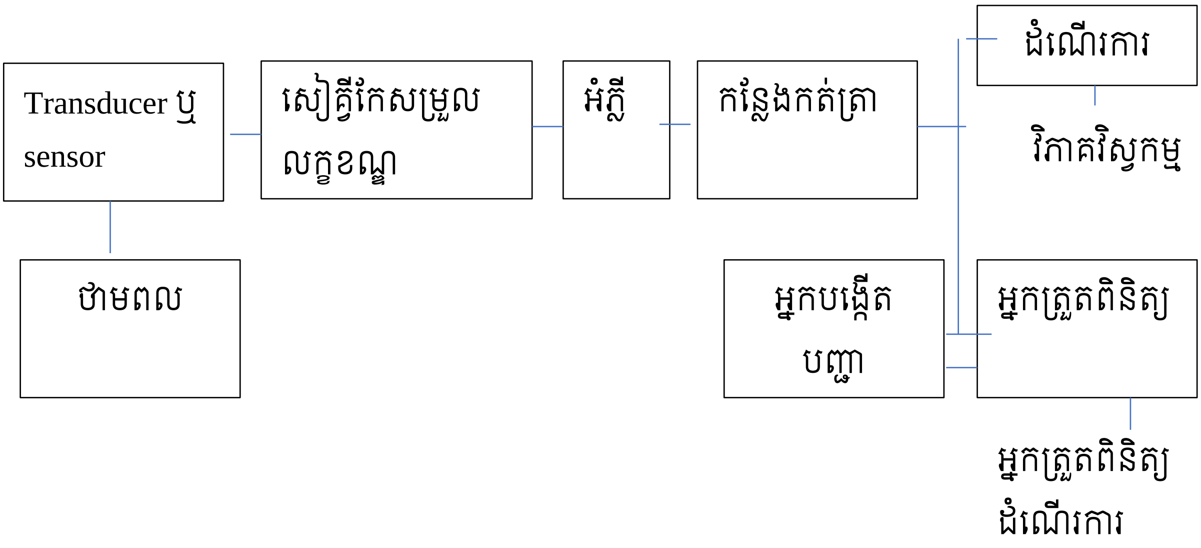
រូបភាព ២.១២ ៖ យ៉ូតបញ្ចេញពន្លឺ

ប្រភព ៖ https://www.explainthatstuff.com

អេក្រង់ Liquid Crystal Display (LCD) ដូចបង្ហាញក្នុងរូបភាព ២.១៣ អាចមានទម្រង់ផ្សេងៗគ្នាដូចជា ៨x១ ១៦x២ ២០x៤ ដែលលេខខាងមុខជាចំនួនអក្សរតាមជួរឈរ និងលេខបន្ទាប់ជាចំនួនអក្សរតាមជួរដេក ឧទាហរណ៍ ៨x១ អាចបង្ហាញអក្សរតាមជួរឈរបាន៨អក្សរ និងមាន១ជួរដេក។

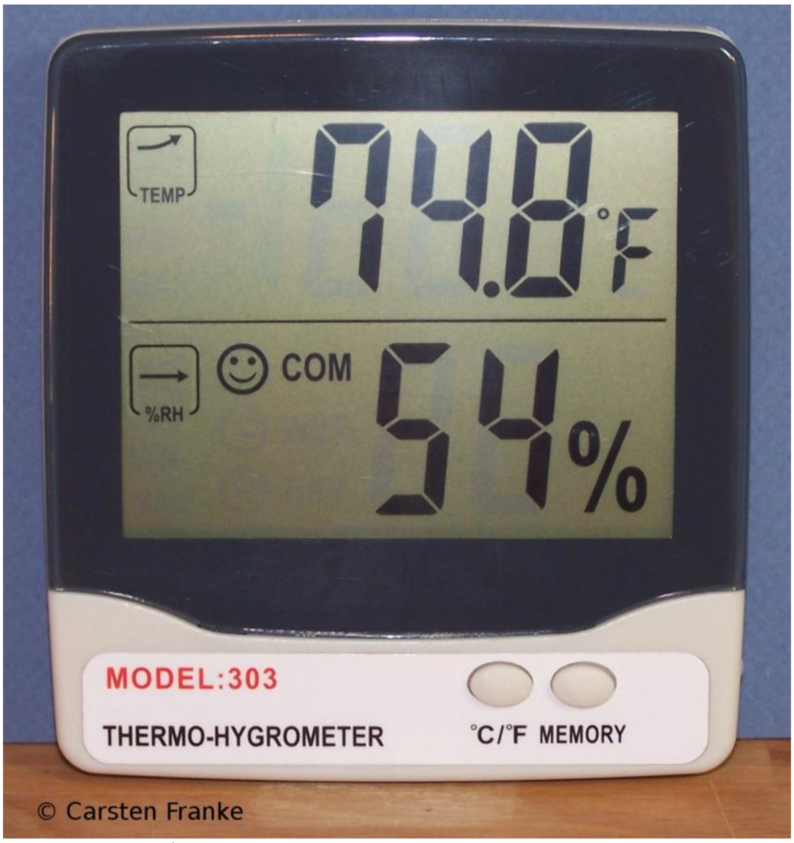
រូបភាព ២.១៣ ៖ អេក្រង់ LCD អាចបង្ហាញអក្សរបាន ១៦ជួរឈរ និង២ជួរដេក

២.២ ការវាស់វែងនិងកត់ត្រាទិន្នន័យ



រូបភាព ២.១៤ ៖ ប្លុកដ្យាក្រាមបង្ហាញពីប្រព័ន្ធអេឡិកត្រូនិកសម្រាប់ការវិភាគវិស្វកម្មនិងត្រួតពិនិត្យដំណើរការ

ក្នុងដំណើរវាស់វែង កត់ត្រាទិន្នន័យ និងត្រួតពិនិត្យដំណើរការ មានប្លុកដ្យាក្រាមដូចបង្ហាញក្នុង រូបភាព ២.១៤ ។ ជាធម្មតាមានប្រភពថាមពលសម្រាប់ដំណើរការ។ Transducer ឬសេនស័រ ជាអ្នកទទួលញាណ ពីប៉ារ៉ាម៉ែត្រដែលត្រូវវាស់វែងដូចជា កំលាំង សម្ពាធ សីតុណ្ណភាព ជាដើម ហើយបង្កើតទៅជាបម្រែបម្រួលតង់ស្យុង ឬអាំងតង់ស៊ីតេចរន្ត ដែលអាចប្រើប្រាស់ជាធាតុចូលសម្រាប់ផ្នែកបន្តទៀតរបស់ប្រព័ន្ធ។ សៀគ្វីកែសម្រួលលក្ខខណ្ឌ ប្រើប្រាស់សម្រាប់ច្រោះសញ្ញារំខានដែលផ្តល់មកពី Transducer ឬសេនស័រ។ សញ្ញាពីសេនស័រមានថាមពលខ្សោយ ក្រោយពីដកសញ្ញារំខានចេញរួចហើយ សញ្ញាត្រូវឆ្លងកាត់អំភ្លីដើម្បីបង្កើនអំភ្លីទុតរបស់សញ្ញាដល់កម្រិតដែលអាចយកទៅប្រើប្រាស់បាន។ ចំណែកឯ កន្លែងកត់ត្រាអាចជាប្រភេទមេកានិក ឬអេឡិកត្រូនិក។ បច្ចុប្បន្ននេះកន្លែងកត់ត្រាអេឡិកត្រូនិកមានប្រើប្រាស់ច្រើន ដូចជា កត់ត្រាចូលក្នុង SD Card ជាដើម។ ទិន្នន័យដែលកត់ត្រារួចហើយ អាចយកទៅដំណើរការទិន្នន័យ បង្កើតជាព័ត៌មានអាចប្រើប្រាស់ក្នុងការវិភាគវិស្វកម្មបាន។ ផ្លូវមួយទៀត ទិន្នន័យទាំងនេះអាចផ្តល់ទៅឲ្យអ្នកត្រួត ពិនិត្យ (controller) ដែលអាចបង្កើតជាបញ្ជាទៅឧបករណ៍បញ្ចេញសកម្មភាព (Actuator) ដូចជា solenoid, relay, ម៉ូទ័រ ជាដើម សម្រាប់កែសម្រួលប៉ារ៉ាម៉ែត្ររបស់ប្រព័ន្ធ។

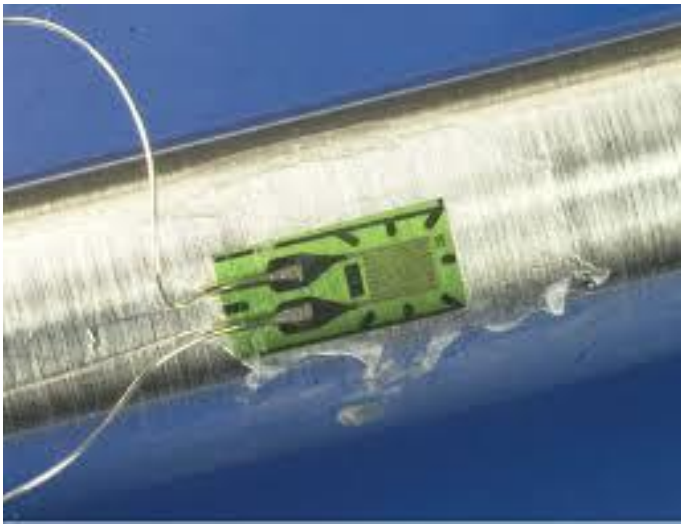


រូបភាព ២.១៥ ៖ ឧបករណ៍វាស់វែងនិងកត់ត្រាសីតុណ្ហភាពនិងសំណើមបរិយាកាសខ្យល់
ប្រភព ៖ Carsten Franke

ឧបករណ៍ក្នុងរូបភាព ២.១៥ ប្រើប្រាស់សម្រាប់វាស់វែង និងកត់ត្រាសីតុណ្ហភាពនិងសំណើម បរិយាសខ្យល់។ ឧបករណ៍នេះផ្តល់ថាមពលដោយថ្មពិល និងមានសេនស័រដែលអាចវាស់សីតុណ្ហភាពនិងសំណើមបរិយាកាសខ្យល់បាន។ នៅខាងក្នុងឧបករណ៍នេះមានសៀគ្វីសម្រាប់កែសម្រួលលក្ខខណ្ឌ អំភ្លី និងកន្លែងកត់ត្រាទិន្នន័យ។ ឧបករណ៍នេះអាចដំណើរការទិន្នន័យ ដើម្បីបង្ហាញសីតុណ្ហភាព និងសំណើមបរិយាកាស នៅលើអេក្រង់ ហើយសីតុណ្ហភាពអាចបង្ហាញជាអង្សាសេ ឬអង្សាហ្វារិនហៃ ដោយចុចលើប៊ូតុង °C/°F។

២.២.១ Transducer ឬសេនស័រ

Transducer ឬសេនស័រជាឧបករណ៍អាចបម្លែងការផ្លាស់ប្តូរ ឬ ប្រែប្រួលលក្ខណៈមេកានិក ឬ បរិមាណកម្តៅទៅជាបរិមាណអគ្គីសនី ឧទាហរណ៍ strain gauge ក្នុងរូបភាព ២.១៦ បម្លែងកំលាំង ឬសម្ពាធទៅជាបរិមាណអគ្គីសនី ចំណែកឯ thermocouple ក្នុងរូបភាព ២.១៧ បម្លែងការប្រែប្រួលសីតុណ្ហភាពទៅជាបរិមាណអគ្គីសនី។



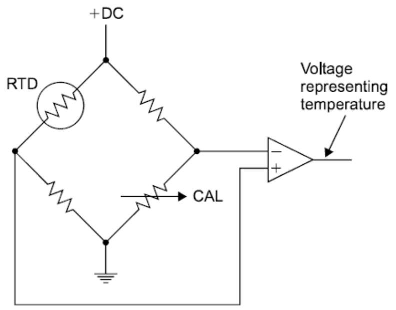
រូបភាព ២.១៦ ៖ Strain gauge

រូបភាព ២.១៧ ៖ Thermocouple

សេនស័រសីតុណ្ហភាពមានច្រើនប្រភេទដូចជា RTD, ទែរមីស្ទ័រ (thermistors), និង thermocouples។ ពួកវាបម្លែងសីតុណ្ហភាពទៅជាការប្រែប្រួលរេស៊ីស្តង់ ឬ តង់ស្យុង។

២.២.១.១ RTD (Resistance Temperature Detector)

វាជាខ្សែផ្លាទីនដែលប្រែប្រួលរ៉េស៊ីស្តង់តាមការប្រែប្រួលសីតុណ្ហភាព។ ភាគច្រើនវាមានរ៉េស៊ីស្តង់ ១០០អូម នៅសីតុណ្ហភាព០អង្សាសេ ហើយពេលសីតុណ្ហភាពកើនឡើង រ៉េស៊ីស្តង់ក៏កើនឡើងដែរ។ ដើម្បីបម្លែងបម្រែបម្រួលរ៉េស៊ីស្តង់ទៅជាបម្រែបម្រួលតង់ស្យុង ជាធម្មតាគេតវានៅក្នុងសៀគ្វីស្ពានដូចក្នុងរូបភាព ២.១៨។

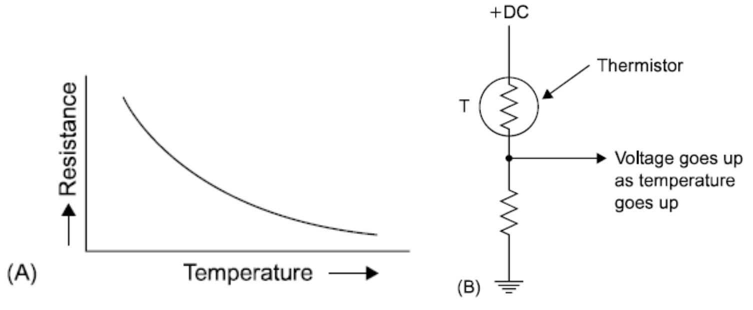


រូបភាព ២.១៨ ៖ ការតភ្ជាប់ RTD ក្នុងសៀគ្វីស្ពាន

២.២.១.២ ទែរមីស្ទ័រ (Thermistors)

ទែរមីស្ទ័រ ដូចក្នុងរូបភាព ២.១៩ ជារេស៊ីស្តង់ដែលតម្លៃរបស់វាថយចុះពេលសីតុណ្ហភាពកើនឡើង។ ការប្រែប្រួលរេស៊ីស្តង់របស់វាធំជាង RTD ច្រើន គេអាចតវាជាមួយសៀគ្វីចែកតង់ស្យុង (voltage divider) ដូចក្នុងរូបភាព ២.២០ (B) ឬ សៀគ្វីស្ពានដូច RTD ក៏បាន។

រូបភាព ២.១៩៖ ទែរមីស្ទ័រ

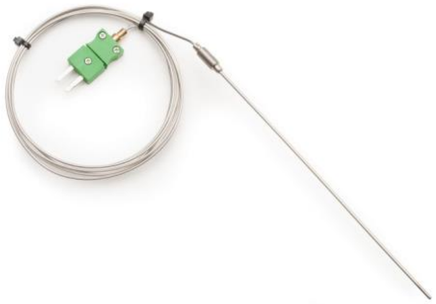


រូបភាព ២.២០ ៖ ទំនាក់ទំនងរវាងសីតុណ្ហភាពនិងរេស៊ីស្តង់របស់ទែរមីស្ទ័រ (A) និងរបៀបតភ្ជាប់ (B)

២.២.១.៣ Thermocouples

Thermocouple នៅក្នុងរូបភាព ២.២១ ជាសេនស័រសីតុណ្ហភាពដែលបង្កើតតង់ស្យុងដោយខ្លួនឯងពេលសីតុណ្ហភាពប្រែប្រួល។ វាបង្កើតដោយលោហៈ២ប្រភេទផ្សេងគ្នា នៅពេលតំណម្ខាងរបស់វាឡើង

សីតុណ្ហភាពវាបង្កើតបានជាតង់ស្យុង ដែលមានតម្លៃតូចជាមីលីវ៉ុល ដែលត្រូវដំឡើងដោយប្រើប្រាស់អំភ្លីសិនមុននឹងអាចយកទៅប្រើប្រាស់បាន។



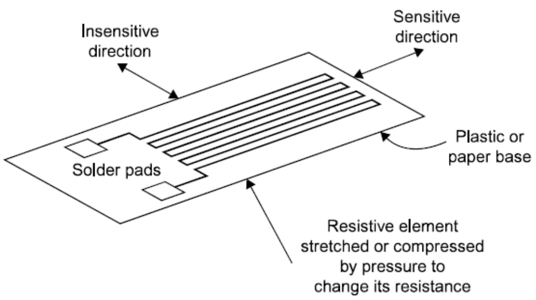
រូបភាព ២.២១ ៖ Thermocouple

២.២.១.៤ សេនស័រសីតុណ្ហភាពប្រភេទ Solid State

គេអាចប្រើប្រាស់ឌីយ៉ូត និងត្រង់ស៊ីស្ទ័រជាសេនស័រសីតុណ្ហភាពបានផងដែរ។ ធម្មតាសេនស័រនេះគេដាក់នៅជាមួយ IC ដែលមានអំភ្លី ហើយវាបង្កើតតង់ស្យុងសមមាត្រនឹងការប្រែប្រួលសីតុណ្ហភាព។

២.២.១.៥ សេនស័រសម្ពាធ

ប្រភេទសេនស័រសម្ពាធដែលគេនិយមប្រើហៅថា strain gauge ដូចក្នុងរូបភាព ២.២២។ រ៉េស៊ីស្តង់របស់វាជាធម្មតាមានតម្លៃ 120 អូម។ គេបិទភ្ជាប់វាដោយការវិទ្យាសាស្ត្រទៅនឹងវត្ថុដែលត្រូវវាស់សំពាធ ឬកំលាំង។ នៅពេលវារងកំលាំង វាយឺត ឬ រួមមាឌ ដែលធ្វើឲ្យរ៉េស៊ីស្តង់របស់វាកើនឡើង។ គេច្រើនតវានៅក្នុងសៀគ្វីស្ពាន ដើម្បីឲ្យការវាស់វែងទទួលបានភាពសុក្រិតខ្ពស់។ បរិមាណអគ្គីសនីចេញពី strain gauge មានតម្លៃតូច ហេតុនេះគេត្រូវប្រើ instrumentation amplifier ដើម្បីបង្កើនតង់ស្យុងរបស់វា។ Load cell ដូចក្នុងរូបភាព ២.២៣ គេច្រើនប្រើប្រាស់ដើម្បីវាស់ទម្ងន់។



រូបភាព ២.២២៖ Strain gauge

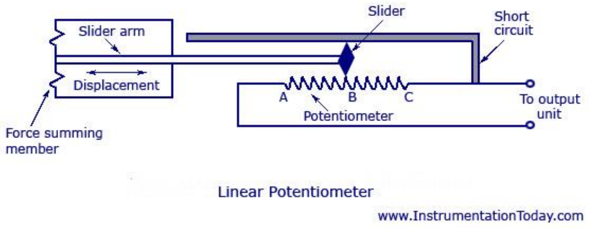
រូបភាព ២.២៣ ៖ Load cell ដែលមានភ្ជាប់ strain gauge ប្រើប្រាស់សម្រាប់វាស់ទម្ងន់

២.២.១.៦ កុងតាក់ (Switches)

កុងតាក់ គេប្រើប្រាស់សម្រាប់ដឹងពីការផ្លាស់ទីដើម្បីដឹងថាវាទៅដល់គោលដៅហើយឬនៅ? ឧទាហរណ៍ កុងតាក់ដើម្បីឲ្យដឹងថាទ្វារបិទ ឬ បើក ឬក៏ដឹងថារបស់អ្វីមួយរត់ដល់ចំណុចណាមួយហើយឬនៅ?

២.២.១.៧ Potentiometers

Potentiometer ដូចក្នុងរូបភាព ២.២៤ ជារេស៊ីស្តង់ប្រែប្រួលតាមប្រវែង ឬ មុំរង្វិល ដែលអាចប្រើប្រាស់សម្រាប់កំណត់ទីតាំង និងមុំ។



រូបភាព ២.២៤៖ Potentiometers

ប្រភព ៖ https://www.instrumentationtoday.com

២.២.១.៨ សេនស័រ Photoelectric

សេនស័រ Photoelectric គឺជាឧបករណ៍ឆ្លើយតបនឹងពន្លឺហើយបង្កើតសញ្ញាអគ្គិសនី។ វាអាចមាន photoconductive ឬ photoresistive, photovoltatic ឬ solar Cell និង ត្រង់ស៊ីស្ទ័រពន្លឺ ដូចបង្ហាញក្នុងរូបភាព ២.២៥។ នៅក្នុងការប្រើប្រាស់ គេអាចតភ្ជាប់វានៅក្នុងសៀគ្វី voltage divider ដូចក្នុងរូបភាព ២.២៦។

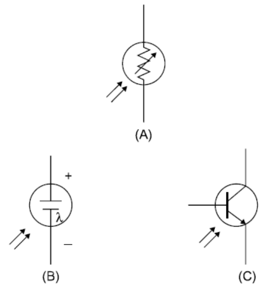
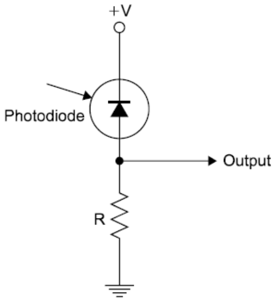


FIGURE 12.9 Schematic symbols for photo cells. (A) Photoconductive or photoresistive. (B) Photovoltaic or solar cell. (C) Photo transistor.

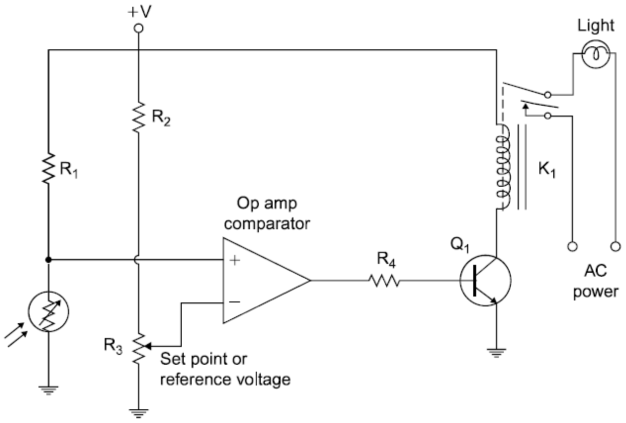
រូបភាព ២.២៥ ៖ សេនស័រ Photoelectric ប្រភេទ Photoconductive ឬ Photoresistive (A) Photovoltatic ឬ Solar Cell (B) ត្រង់ស៊ីស្ទ័រពន្លឺ (C)



រូបភាព ២.២៦ ៖ ការតភ្ជាប់ឌីយ៉ូតពន្លឺក្នុងសៀគ្វី voltage divider

សៀគ្វីបិទបើកភ្លើងដោយពន្លឺ

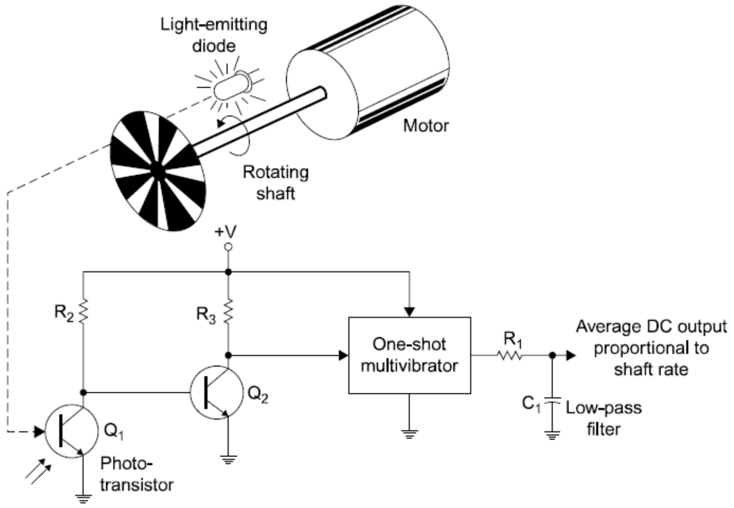
សៀគ្វីនេះដូចបង្ហាញនៅក្នុងរូបភាព ២.២៧ ប្រើប្រាស់ Photoconductive ឬ Photoresistive ដើម្បីបើកភ្លើងដោយស្វ័យប្រវត្តិនៅពេលងងឹត។ នៅពេលគ្មានពន្លឺ ឬ ពន្លឺមិនគ្រប់គ្រាន់ រ៉េស៊ីស្តង់របស់ Photoresistive ក៏ប្រែប្រួល។



រូបភាព ២.២៧ ៖ ការប្រើប្រាស់សេនស័រ photoconductive ដើម្បីបិទបើកអំពូលភ្លើង

ឧបករណ៍វាស់ល្បឿនជុំប្រើប្រាស់ហ្វូតូត្រង់ស៊ីស្ទ័រ

ឧបករណ៍នេះប្រើប្រាស់សេនស័រ photoelectric ប្រភេទត្រង់ស៊ីស្ទ័រពន្លឺ ដែលទទួលពន្លឺពី LED ដែលឆ្លងកាត់ឌីសមានចោះរន្ធ ពន្លឺដែលត្រង់ស៊ីស្ទ័រពន្លឺទទួលបាននៅលើ base របស់វាលឿន ឬ យឺតអាស្រ័យទៅលើល្បឿនជុំរបស់ម៉ូទ័រ ដែលបង្កើតចរន្តជាប់មានតម្លៃមធ្យមសមមាត្រទៅនឹងល្បឿនជុំភ្លៅរបស់ម៉ូទ័រ។



រូបភាព ២.២៧ ៖ ការប្រើប្រាស់ត្រង់ស៊ីស្ទ័រពន្លឺក្នុងឧបករណ៍វាស់ល្បឿនជុំ

២.២.១.៩ សេនស័រលំហូរ

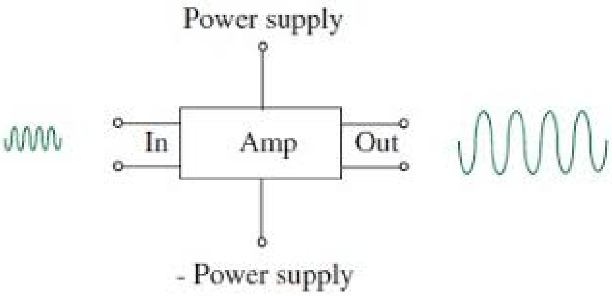
សេនស័រលំហូរបង្កើតតង់ស្យុងសមមាត្រទៅនឹងបរិមាណលំហូរអង្គធាតុរាវ ឬ ខ្យល់។ គេប្រើប្រាស់វាស់ដើម្បីវាស់វែងរបស់លំហូររបស់ទឹកហូរក្នុងបំពង់ ឬ វាស់បរិមាណជីវឧស្ម័នហូរក្នុងបំពង់បញ្ជូនជីវឧស្ម័ន។

២.២.២ សៀគ្វីកែសម្រួលលក្ខខណ្ឌ (Signal Conditioner)

សៀគ្វីកែសម្រួលលក្ខខណ្ឌប្រើសម្រាប់បម្លែង ទូទាត់គ្នា ឬសម្រាប់សម្របសម្រួលផលបានមកពី transducer ឬ sensor វាមានដូចជា តម្រង (filter), compensator, modulator, integrator។ល។

២.២.៣ អំភ្លី (amplifier)

អំភ្លីជំឡើងផលអគ្គីសនីចេញពី transducer ដែលមានតង់ស្យុងទាបឲ្យកើនឡើងដល់កម្រិតមួយអាចយកទៅប្រើប្រាស់ជាមួយឧបករណ៍វាស់តង់ស្យុងបាន មើលរូបភាព ២.២៨។



រូបភាព ២.២៨ ៖ អំភ្លី (A) បង្ហាញសញ្ញាដែលចូលនិងចេញពីអំភ្លី (B)

២.២.៤ ឧបករណ៍កត់ត្រា

ឧបករណ៍កត់ត្រាមានដូចជា multimeter, oscilloscope, កាសែតថតសម្លេង និង SD card ជាដើម។ Multimeter បង្ហាញនៅក្នុងរូបភាព ២.២៩ អាចប្រើប្រាស់សម្រាប់វាស់វែងតង់ស្យុង អាំងតង់ស៊ីតេ រ៉េស៊ីស្តង់។ល។

រូបភាព ២.២៩ ៖ Multimeter

Oscilloscope ក្នុងរូបភាព ២.៣០ អាចប្រើប្រាស់សម្រាប់វាស់សញ្ញាអគ្គីសនី ដែលអាចពិនិត្យមើលការប្រែប្រួលតម្លៃរបស់វាតាមពេលវេលា គេអាចវាស់វែងដឹងតង់ស្យុង ហ្វ្រេកង់។

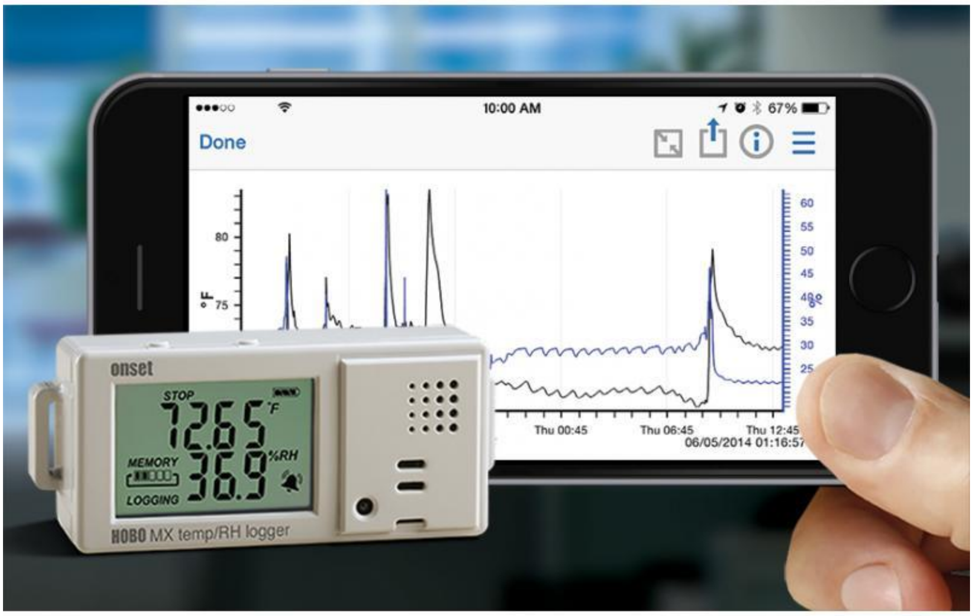
រូបភាព ២.៣០ ៖ Oscilloscope

SD card កំពុងពេញនិយមប្រើប្រាស់ក្នុងការកត់ត្រានិងរក្សាទុកទិន្នន័យនៅក្នុងឧបករណ៍វាស់វែងនិងកត់ត្រាទិន្នន័យ កុំព្យូទ័រ និង Smart phone។ល។ ដូចបង្ហាញក្នុងរូបភាព ២.៣១។ Mico SD card មានប្រើប្រាស់ច្រើននៅក្នុង smart phone ដោយសារទំហំរបស់វាតូច គេអាចប្រើប្រាស់វាជាមួយ SD card reader បានដែរ ដោយប្រើប្រាស់ SD card adapter។

រូបភាព ២.៣១ ៖ SD Card

២.២.៥ ដំណើរការទិន្នន័យ (Data Processor)

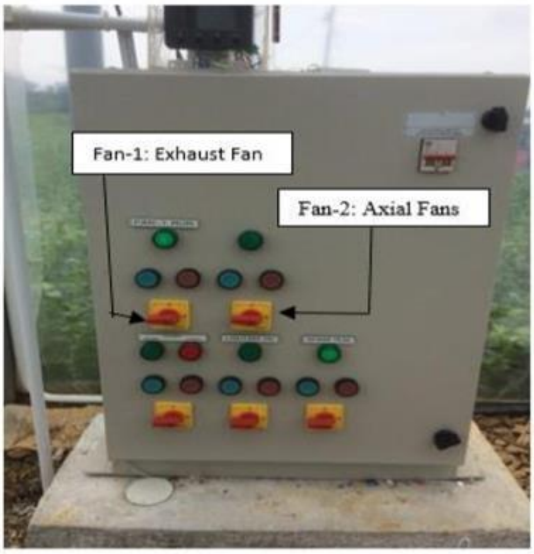
ដំណើរការទិន្នន័យ (Data Processor) បម្លែងសញ្ញាចេញពីឧបករណ៍វាស់ទៅជាទិន្នន័យដែលអាចបកស្រាយដោយក្រុមវិស្វករបាន ដោយធ្វើការសង្ខេបទិន្នន័យ និងបង្ហាញទិន្នន័យជាក្រាហ្វិកជាដើម ដូចក្នុងរូបភាព ២.៣២។



រូបភាព ២.៣២ ៖ ដំណើរការទិន្នន័យ (Data processor)

២.២.៦ អ្នកត្រួតពិនិត្យដំណើរការ (Process Controller)

វាបង្ហាញព័ត៌មានពីដំណើរការ ហើយធ្វើការកែសម្រួលបរិមាណមេកានិក ឬ កម្តៅនៅក្នុងដំណើរការណាមួយ។ នៅក្នុងរូបភាព ២.៣៣ ជាអ្នកត្រួតពិនិត្យដំណើរការនៅក្នុងផ្ទះសំណាញ់ ដែលមានអំពូលភ្លើងច្រើនប្រភេទអាចប្រាប់អ្នកប្រើប្រាស់ឲ្យដឹងពីស្ថានភាពរបស់ឧបករណ៍ដែលត្រូវបញ្ហា ដូចជា កង្ហារកំពុងដំណើរការ ឬ កំពុងឈប់ ជាដើម។ ប្រភេទប្រព័ន្ធវាអាចដំណើរការដោយមនុស្ស ឬ អាចដំណើរការដោយស្វ័យប្រវត្តិ អាស្រ័យលើព័ត៌មានពីសេនស័រដែលដាក់នៅក្នុងផ្ទះសំណាញ់។



រូបភាព ២.៣៣ ៖ អ្នកត្រួតពិនិត្យដំណើរការ

ឧទាហរណ៍ ឧបករណ៍បញ្ជាសីតុណ្ហភាព មាន sensor វាស់សីតុណ្ហភាពប្រើប្រាស់រ៉េស៊ីស្តង់ដើម្បីតាមដានសីតុណ្ហភាពរបស់អង្គធាតុរាវនៅក្នុងបំពង់ទឹក ពេលសីតុណ្ហភាពកើនឡើងសញ្ញាចេញពី sensor ក៏កើនឡើងដែរ។ នៅក្នុងឧបករណ៍បញ្ជាមាន ឧបករណ៍ប្រៀបធៀប (comparator) ដែលគេបានកំណត់សីតុណ្ហភាពឲ្យវាបញ្ជាបើក ឬ បិទ នៅពេលសីតុណ្ហភាពកើនឡើងហួសកម្រិតកំណត់ វានឹងបញ្ជាឲ្យឧបករណ៍កម្តៅឈប់ នៅពេលសីតុណ្ហភាពអង្គធាតុរាវធ្លាក់ចុះដល់កម្រិតកំណត់វាបញ្ជាឲ្យដំណើរការឡើងវិញ។ បច្ចុប្បន្ននេះ ឧបករណ៍បញ្ជាឌីជីថលមានកុំព្យូទ័រនៅក្នុងនោះហៅថា Programmable Logic Controller (PLC) ឬ ប្រើប្រាស់ microconstroller។

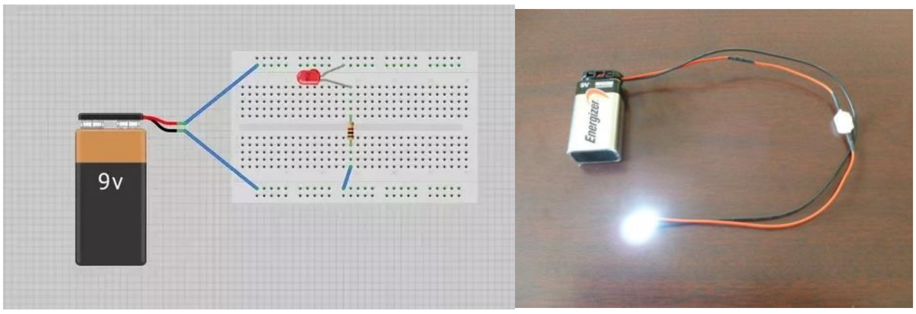
២.៣ ការវិភាគវិស្វកម្ម (Engineering Analysis)

ការវិភាគវិស្វកម្មអាចធ្វើតាមពីរប្រភេទគឺ ដោះស្រាយតាមទ្រឹស្តីនិងដោះស្រាយតាមការពិសោធន៍។

ដំណោះស្រាយតាមទ្រឹស្តី (theoretical approach) ប្រើប្រាស់សមីការសរសេរឡើងដោយពណ៌នាអំពីឥរិយាបថរបស់គំរូ (model) ដែលត្រូវធ្វើការវិភាគ ឧទាហរណ៍ ប្រើប្រាស់ simulator ដើម្បីធ្វើតេស្ត។ នៅក្នុងរូបភាព ២.៣៤ (A) ប្រើប្រាស់ simulator ដើម្បីមើលសៀគ្វីប្រើប្រាស់ថ្មពិល ៩វ៉ុល ដើម្បីឲ្យ LED ដំណើរការ។ ជាធម្មតា បើក្នុង simulator ដំណើរការបានល្អ នៅពេលផលិតជា protype ដាក់ស្តែងក៏ដំណើរការស្រដៀងនេះដែរ។

ដំណោះស្រាយតាមការពិសោធន៍ ធ្វើបានតាមរយៈការផលិតជា prototype ហើយយកទៅធ្វើតេស្តដាក់ស្តែងតែម្តង និងវាស់វែងប៉ារ៉ាម៉ែត្រដែលចង់ផ្ទៀងផ្ទាត់។ នៅក្នុងរូបភាព ២.៣៤ (B) គឺជាការពិសោធន៍

ដាក់ស្តែងទៅលើសៀគ្វីខាងលើ។ ជាធម្មតា ការពិសោធន៍ដាក់ស្តែងចំណាយធនធានអស់ច្រើនជាងវិភាគតាមទ្រឹស្តី។ ហេតុនេះហើយ បើអាចមុននឹងធ្វើការពិសោធន៍ដាក់ស្តែង គេច្រើនធ្វើការវិភាគតាមទ្រឹស្តីជាមុនសិន។



(A)

(B)

រូបភាព ២.៣៤ ៖ ការវិភាគវិស្វកម្ម តាមទ្រឹស្តី (A) និងតាមការពិសោធន៍លើ prototype (B)

២.៤ ប្រព័ន្ធបញ្ជាស្វ័យប្រវត្តិ

ប្រព័ន្ធបញ្ជាស្វ័យប្រវត្តិត្រូវបានគេប្រើប្រាស់ក្នុងឧស្សាហកម្ម (Industrial Control) ដើម្បីត្រួតពិនិត្យ និងបញ្ជាខ្សែច្រវាក់ផលិតកម្ម។ ចំណែកនៅក្នុងវិស័យកសិកម្មកសិកម្ម ក៏មានប្រើប្រាស់នៅក្នុងរោងចក្រផលិតម៉ាស៊ីន ឬ ឧបករណ៍កសិកម្ម រោងចក្រកែច្នៃផលិតផលកសិកម្ម និងនៅក្នុងកន្លែងបញ្ជាគ្រឿងយន្ត ឬ ម៉ាស៊ីនកសិកម្ម។ល។ ដោយប្រើប្រាស់គ្រឿងអេឡិកត្រូនិក សៀគ្វី និងឧបករណ៍។ ប្រព័ន្ធបញ្ជាស្វ័យប្រវត្តិដំណើរការជាលក្ខណៈម៉េកានិក ឬ បញ្ជាដោយអគ្គីសនី គេប្រើប្រាស់វាដើម្បីបើក ឬ បិទនៅពេលវេលាសមស្រប បញ្ជាល្បឿនដំណើរការ និងធ្វើការកែសម្រួលដំណើរការ។ ផលប្រយោជន៍នៃប្រព័ន្ធបញ្ជាដោយស្វ័យប្រវត្តិមាន ដូចជា បង្កើនផលិតភាពការងារ និងកាត់បន្ថយគ្រោះថ្នាក់របស់បុគ្គលិក ដោយសារការងារធ្វើដោយម៉ាស៊ីនស្វ័យប្រវត្តិ ឬមនុស្សយន្ត (robot)។

២.៤.១ ឧបករណ៍សម្រាប់ត្រួតពិនិត្យដំណើរការ

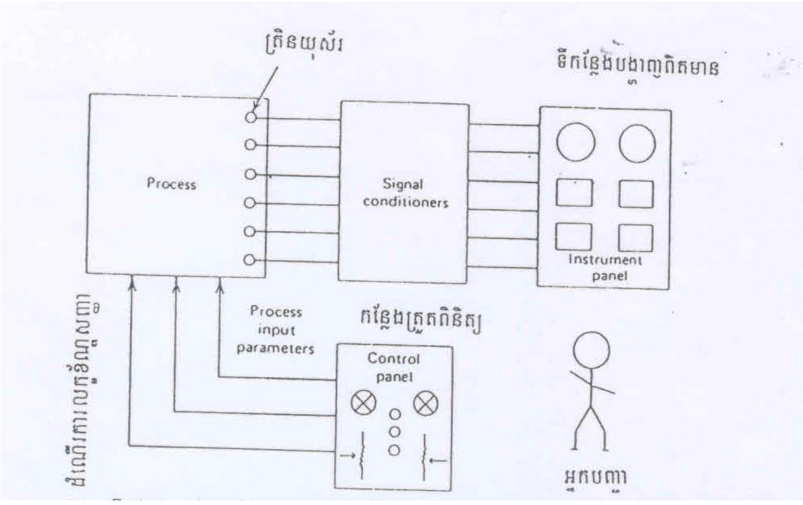
ការត្រួតពិនិត្យត្រូវការចាំបាច់ដើម្បីឲ្យដំណើរប្រព្រឹត្តទៅដូចដែលយើងចង់បាន ហេតុនេះហើយត្រូវការកែសម្រួលជារឿយៗ បន្ទាប់ពីបានទទួលព័ត៌មានពី transducer ឬ sensor ។ ឧទាហរណ៍ ស៊ីទែនទឹកមានឧបករណ៍សម្រាប់វាស់កម្ពស់ទឹកនិងបិទបើកម៉ូទ័របូមទឹកដោយស្វ័យប្រវត្តិ។ នៅពេលម៉ូទ័របូមទឹកពេញដល់កម្រិតដែលបានកំណត់ជាមុន ឧបករណ៍វាស់កម្ពស់ទឹកនឹងផ្តាច់ចរន្តអគ្គីសនីដើម្បីឲ្យម៉ូទ័រឈប់ដំណើរការ។ នៅពេលយើងប្រើប្រាស់ទឹក ហើយកម្ពស់ទឹកក្នុងស៊ីទែនថយចុះដល់កម្រិតមួយ វានឹងភ្ជាប់ចរន្ត

ទៅម៉ូទ័រវិញ ដើម្បីចាប់ផ្ដើមបូមទឹកចូលក្នុងស៊ីទែនបន្ថែម។ ប្រព័ន្ធបញ្ជាស្វ័យប្រវត្តិនេះ ធានាថានៅក្នុងស៊ីទែនមានទឹកជាប្រចាំ ដោយមិនត្រូវការមនុស្សទៅបើក ឬ បិទ ម៉ូទ័របូមទឹកទេ។

២.៤.១.១ ប្រព័ន្ធបញ្ជា Open-loop និង Closed-loop

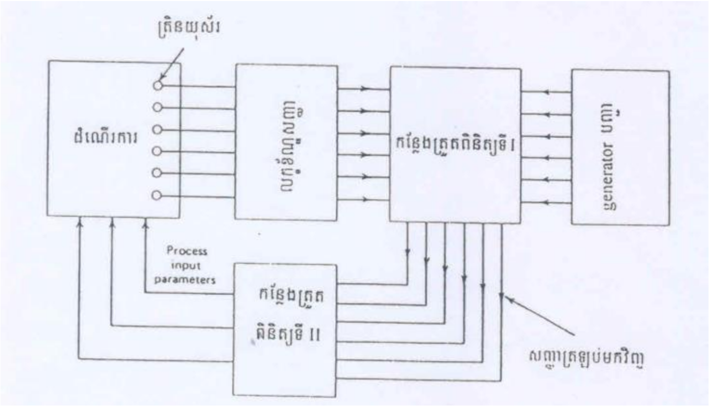
ប្រព័ន្ធបញ្ជាអាចចែកជាពីរប្រភេទគឺ Open-loop Control (ត្រូវការមនុស្សដើម្បីតាមដាននិងបញ្ជា) និងប្រព័ន្ធ Closed-loop Control (ដំណើរការដោយស្វ័យប្រវត្តិ និងត្រូវការអន្តរាគមន៍ពីមនុស្សម្ដងម្កាល)។

ប្រព័ន្ធ Open-loop Control មានប្លុកដ្យាក្រាមដូចក្នុងរូបភាព ២.៣៥។ នៅក្នុងប្រព័ន្ធនេះ មនុស្សអាចបញ្ជានិងកែសម្រួលដំណើរការដោយប្រើប្រាស់កន្លែងត្រួតពិនិត្យ។ ត្រិនយុស័រវាស់វែងនិងតាមដានដំណើរការ ហើយបញ្ជូនទិន្នន័យនិងព័ត៌មានឆ្លងកាត់អ្នកកែសម្រួលលក្ខខណ្ឌ (signal conditioners) រួចបង្ហាញព័ត៌មានដល់អ្នកបញ្ជានៅទីកន្លែងបង្ហាញព័ត៌មាន។ ពីព័ត៌មានដែលបានធ្វើបច្ចុប្បន្នភាពនេះ អ្នកបញ្ជាអាចសម្រេចចិត្តបញ្ជាដំណើរការតាមរយៈផ្ទាំងបញ្ជា (control panel)។



រូបភាព ២.៣៥ ៖ ដ្យាក្រាមប្រព័ន្ធបញ្ជា open-loop ដែលត្រូវការមនុស្សបញ្ជានិងកែសម្រួលប៉ារ៉ាម៉ែត្រដំណើរការ

ចំណែកឯប្រព័ន្ធបញ្ជា Closed-loop Control ដូចក្នុងរូបភាព ២.៣៦ វិញ ដំណើរការដោយស្វ័យប្រវត្តិ មិនមានប្រើប្រាស់មនុស្សទេ។ មនុស្សត្រូវជំនួសដោយ អ្នកបញ្ជា (controller) ដែលអាចជាកុំព្យូទ័រ ដូចជា microcontroller ឬ programmable logic controller (PLC) ដោយបញ្ជូនសញ្ញាត្រលប់ទៅកន្លែងដំណើរការវិញ។



រូបភាព ២.៣៦ ៖ ប្លុកដ្យាក្រាមប្រព័ន្ធបញ្ជា closed-loop

២.៤.២ ឧបករណ៍សម្រាប់កែសម្រួលប៉ារ៉ាម៉ែត្រ

នៅក្នុងប្រព័ន្ធបញ្ជា ឧបករណ៍សម្រាប់កែសម្រួលប៉ារ៉ាម៉ែត្រ ជាអ្នកធ្វើសកម្មភាព ឬ កែសម្រួល ប៉ារ៉ាម៉ែត្រ មានឧបករណ៍ដូចខាងក្រោម ៖

- ម៉ូទ័រចរន្តជាប់
- Stepping Motors
- Solenoids
- ស៊ូប៉ាប់បើកបិទដោយម៉ូទ័រ (motorized valves)
- Servo Control Valve
- ឧបករណ៍ផ្លាស់ប្តូរទីតាំង
- រ៉េស៊ីស្តង់កម្តៅ

២.៤.២.១ ឧបករណ៍បញ្ចេញសកម្មភាព (Actuator)

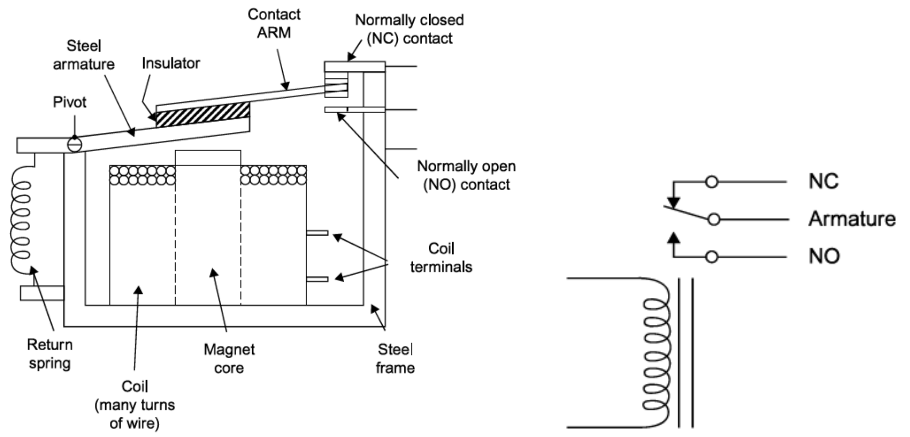
ជាឧបករណ៍កែសម្រួលប៉ារ៉ាម៉ែត្រដែលអាចកែសម្រួលដោយផ្ទាល់ ឬ ដោយប្រយោល (បញ្ជាឧបករណ៍មួយទៀត មុននឹងអាចកែសម្រួលប៉ារ៉ាម៉ែត្រណាមួយបាន ដូចជា relay ជាដើម)។ ប្រភេទសំខាន់ៗរបស់វាមាន ៖

- Relay
- Solenoid
- ម៉ូទ័រ (ចរន្តជាប់ និង ចរន្តឆ្លាស់)

Relay

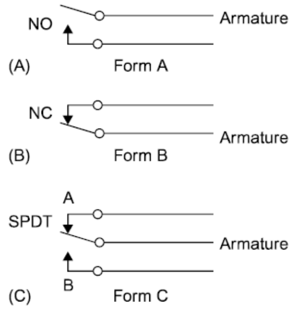
វាជាកុងតាក់ដំណើរការដោយអេឡិកត្រូម៉ាញ៉េទិក អាចដំណើរការដោយចរន្តជាប់និងចរន្តឆ្លាស់ ដូចបង្ហាញនៅក្នុងរូបភាព ២.៣៧។ គេប្រើប្រាស់វាដើម្បីបញ្ជាចរន្តមានតង់ស្យុងខ្ពស់ ដោយប្រើប្រាស់ចរន្តមានតង់ស្យុងទាបបើក ឬ បិទ ទៅក្នុង coil របស់ relay តាមរយៈកុងតាក់។ ការបញ្ជាបើកបិទចរន្តមានតង់ស្យុង

ខ្ពស់ ដោយផ្ទាល់ប្រឈមនឹងគ្រោះថ្នាក់ឆក់ចរន្តអគ្គីសនីដល់អ្នកបញ្ជា ហើយអាចធ្វើឲ្យបែកផ្កាភ្លើងខ្លាំងនៅកន្លែងកុងតាក់បិទបើក ដែលអាចបង្កគ្រោះថ្នាក់ និងឆាប់ឆេះខូចកុងតាក់។ ផ្នែកខាងតង់ស្យុងខ្ពស់មានខ្សែចំនួន៣ Armature ឬ Common ជាខ្សែរួម Normally Open (NO) ខ្សែជាសៀគ្វីបើក នៅពេល coil របស់ relay មិនទាន់ដំណើរការ ចំណែកឯ Normally Closed (NC) ជាខ្សែសៀគ្វីបិទនៅពេល coil របស់ relay មិនទាន់ដំណើរការ។ NO និង NC នឹងត្រលប់ស្ថានភាព នៅពេល coil របស់ relay ដំណើរការ។ ឧទាហរណ៍ អំពូលភ្លើងដែលតភ្ជាប់ទៅនឹង NO នឹងមិនភ្លឺទេ នៅពេល coil របស់ relay មិនទាន់ដំណើរការ វានឹងភ្លឺនៅពេល coil របស់ relay ដំណើរការ។ អំពូលភ្លើងនេះនឹងដំណើរការបញ្ច្រាសវិញ ប្រសិនបើវាតភ្ជាប់ទៅនឹង NC។



រូបភាព ២.៣៧ ៖ ផ្នែកសំខាន់ៗរបស់ Relay (A) និមិត្តសញ្ញាតំណាងឲ្យ Relay (B)

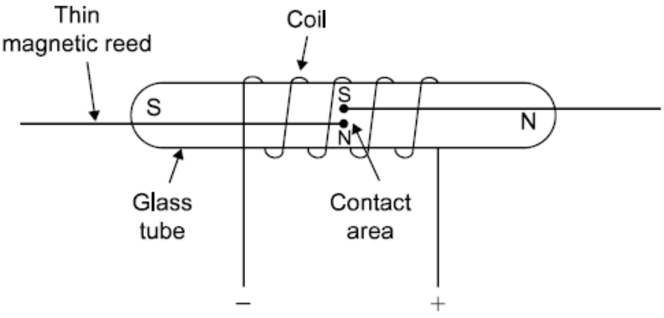
ប្រភេទ Relay អាចមាន៣ទម្រង់ដូចក្នុងរូបភាព ២.៣៨។ (A) NO មិនដំណើរការពេល coil របស់ relay មិនដំណើរការ (B) ដំណើរការពេល coil របស់ relay មិនដំណើរការ និង (C) បញ្ចូលគ្នាប្រភេទទាំងពីរ។ SPDT មកពីពាក្យពេញ Single Pole Double Throw។



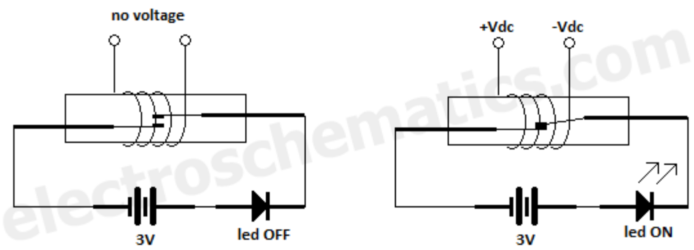
រូបភាព ២.៣៨ ៖ ទម្រង់ទាំងបីប្រភេទរបស់ relay

Reed Relay

ជាប្រភេទ relay ដែលប្រើប្រាស់បន្ទះលោហៈស្តើងរស់ពីរសន្លឹក ជាធម្មតាពួកវាដាច់ពីគ្នា ដូចក្នុង រូបភាព ២.៣៩។ នៅពេល coil ដំណើរការ ដែនម៉ាញ៉េទិកបានធ្វើឲ្យបន្ទះលោហៈទាំងពីរនេះក្លាយជាមេដែក ម្ខាងមានប៉ូល N និងម្ខាងទៀតមានប៉ូល S ធ្វើឲ្យវាធក់ជាប់គ្នាអាចឲ្យចរន្តឆ្លងកាត់បាន។ នៅក្នុងរូបភាព ២.៤០ LED ភ្លឺ នៅពេលបន្ទះលោហៈទាំងពីរធក់ជាប់គ្នា។

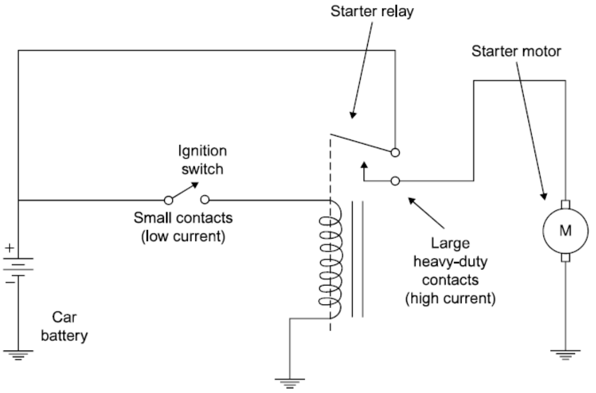


រូបភាព ២.៣៩ ៖ Reed relay



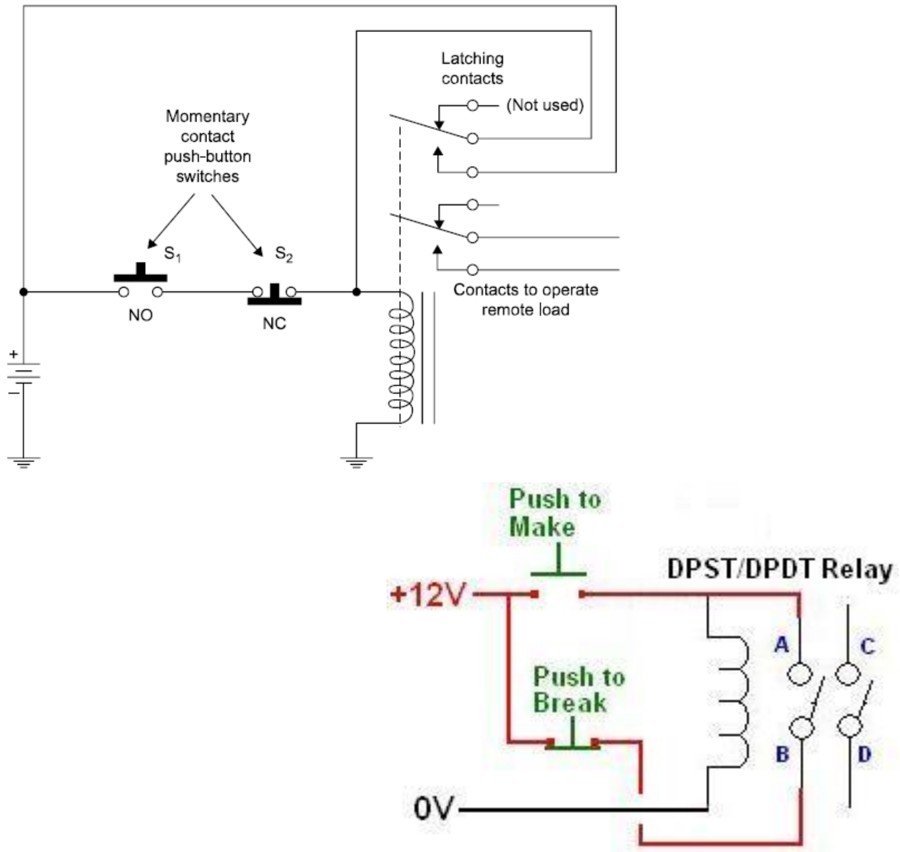
រូបភាព ២.៤០ ៖ Reed relay តភ្ជាប់ជាមួយ LED

នៅក្នុងយានយន្ត គេប្រើ Relay ដើម្បីបញ្ជាចរន្តអគ្គីសនីធំ ពីអាគុយដើម្បីអូសទាញម៉ូទ័រដេម៉ារ័ ក្នុងបង្ហាញក្នុងរូបភាព ២.៤១ នៅពេលកាច់សោរបញ្ឆេះឡាន។ នៅពេលពេលគេកាច់សោរបញ្ឆេះយានយន្តកុងតាក់សោរបានបើកចរន្តអគ្គីសនីពីអាគុយឲ្យឆ្លងកាត់ coil របស់ relay ធ្វើឲ្យវាតភ្ជាប់ចរន្តអគ្គីសនីពីអាគុយទៅម៉ូទ័រដេម៉ារ័តាមរយៈខ្សែចម្លងអគ្គីសនីមួយផ្សេងទៀតដែលមានទំហំធំអាចទប់ទល់នឹងអាំងតង់ស៊ីតេចរន្តខ្ពស់ដែលត្រូវការដោយម៉ូទ័រដេម៉ារ័។ ប្រសិនបើមិនប្រើប្រាស់ relay ហើយតភ្ជាប់ម៉ូទ័រដេម៉ារ័ដោយផ្ទាល់ទៅនឹងអាគុយតែម្តងនឹងមានការបែកផ្កាភ្លើងខ្លាំងកន្លែងប៉ះគ្នា អាចបង្កគ្រោះថ្នាក់ដល់ប្រតិបត្តិករ បង្កជាអគ្គីភ័យ ឬ ធ្វើឲ្យអាគុយរបស់យានយន្តធាប់ខូច។



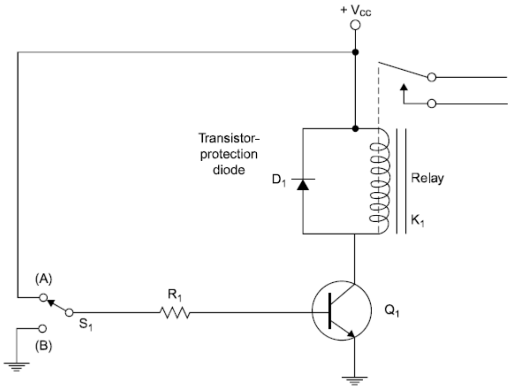
រូបភាព ២.៤១ ៖ ប្រព័ន្ធដេម៉ារ័យានយន្ត

ប្រព័ន្ធបញ្ជា Relay ប្រើប្រាស់កុងតាក់ពីរ (បិទ បើក) ដូចបង្ហាញនៅក្នុងរូបភាព ២.៤២។ ក្នុងការបញ្ជាម៉ូទ័រក្នុងរោងចក្រ ឬ ម៉ាស៊ីនមួយចំនួន ច្រើនប្រើប្រាស់ប្រព័ន្ធប្រភេទនេះ ដោយមានកុងតាក់មួយសម្រាប់បើក និងកុងតាក់មួយទៀតសម្រាប់បិទ។ នៅពេលគេចុចកុងតាក់ S1 ចរន្តពីជនិតា (អាគុយ ឬ ដុំភ្លើង) រត់ឆ្លងកាត់ coil របស់ relay វាទាញប៉ូលខាងលើរបស់ relay បើកឲ្យចរន្តអគ្គីសនីរត់កាត់ coil ទោះបីជាប្រតិបត្តិករឈប់ ចុចកុងតាក់ S1 ក៏ដោយ។ ចំណែកប៉ូលខាងក្រោម ដំណើរការបន្ទុកដែលនៅឆ្ងាយពី relay នេះ។ Relay វាឈប់ដំណើរការ រហូតដល់ប្រតិបត្តិករចុចកុងតាក់ S2។



រូបភាព ២.៤២ ៖ ប្រព័ន្ធបញ្ជា Relay ប្រើប្រាស់កុងតាក់ពីរ

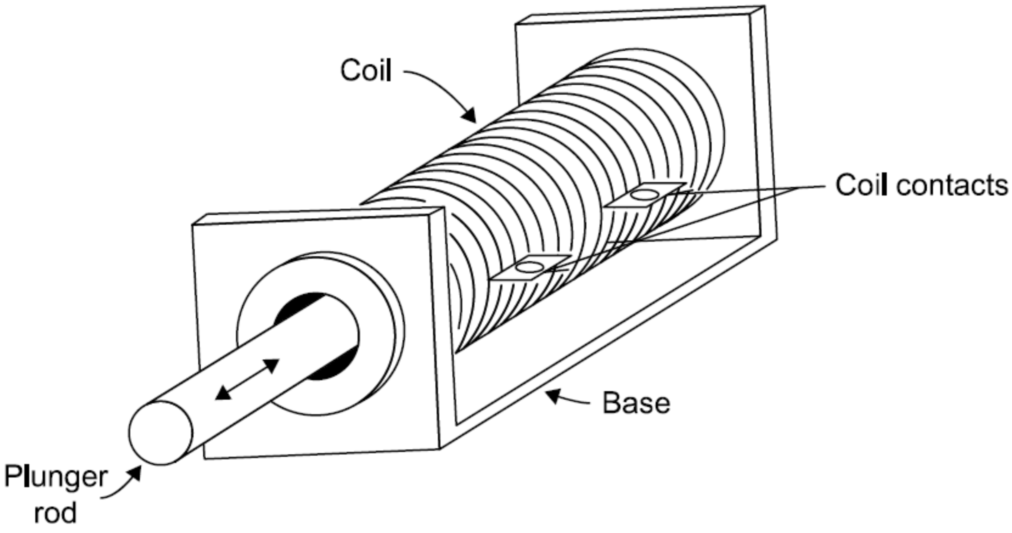
នៅក្នុង relay module សម្រាប់ប្រើប្រាស់ជាមួយ microcontroller ដូចជា Arduino ជាដើម គេច្រើនដំណើរការ Relay ដោយប្រើប្រាស់ត្រង់ស៊ីស្ទ័រដូចក្នុងរូបភាព ២.៤៣។ នៅពេលគេបើកកុងតាក់ S1 ទៅតភ្ជាប់នឹងចំណុច (A) ចរន្តអគ្គីសនីរត់ពី +Vcc ទៅ base របស់ត្រង់ស៊ីស្ទ័រ Q1 ដោយឆ្លងកាត់រ៉េស៊ីស្តង់ (R1 ដែលគេដាក់ដើម្បីបញ្ចុះតង់ស្យុងរបស់ +Vcc) ធ្វើឲ្យត្រង់ស៊ីស្ទ័រចម្លងចរន្តអគ្គីសនីពី +Vcc ឆ្លងកាត់ coil របស់ relay ត្រង់ស៊ីស្ទ័រ Q1 រត់ទៅកាន់ម៉ាស់ ធ្វើឲ្យ relay ដំណើរការ។ នៅពេលគេបើកកុងតាក់ S1 ទៅតភ្ជាប់នឹងចំណុច (B) វិញ base របស់ត្រង់ស៊ីស្ទ័រ Q1 ត្រូវបានតភ្ជាប់ទៅនឹងម៉ាស់ដែលមានប៉ូតង់ស្យែល 0វ៉ុល ធ្វើឲ្យត្រង់ស៊ីស្ទ័រឈប់ចម្លងចរន្តអគ្គីសនីពី +Vcc ឆ្លងកាត់ coil របស់ relay ត្រង់ស៊ីស្ទ័រ Q1 រត់ទៅកាន់ម៉ាស់ ធ្វើឲ្យ relay ឈប់ដំណើរការដែរ។ នៅពេលបិទ relay នេះ វាបង្កើតជាចរន្តមានតង់ស្យុងខ្ពស់រយៈពេលខ្លីដោយសារការប្រែប្រួលភ្លុចម៉ាញ៉េទិកដ៏លឿននៅក្នុង coil របស់ relay ដែលចរន្តនេះអាចធ្វើឲ្យត្រង់ស៊ីស្ទ័រ Q1 ឆេះខូចបាន ហេតុនេះគេតភ្ជាប់ឌីយ៉ូតជាខ្នែងជាមួយ coil របស់ relay ដើម្បីទប់ចរន្តនេះ។



រូបភាព ២.៤៣ ដំណើរការ relay ដោយប្រើប្រាស់ត្រង់ស៊ីស្ទ័រ

Solenoid

Solenoid ជាអេឡិកត្រូម៉ាញ៉េទិកដែលអាចប្រើប្រាស់សម្រាប់បិទបើកវ៉ាន គន្លឹះសោរ ដូចបង្ហាញក្នុងរូបភាព ២.៤៤ វាផ្គុំឡើងដោយ coil សំបក និងប្លុងស័រ (armature)។

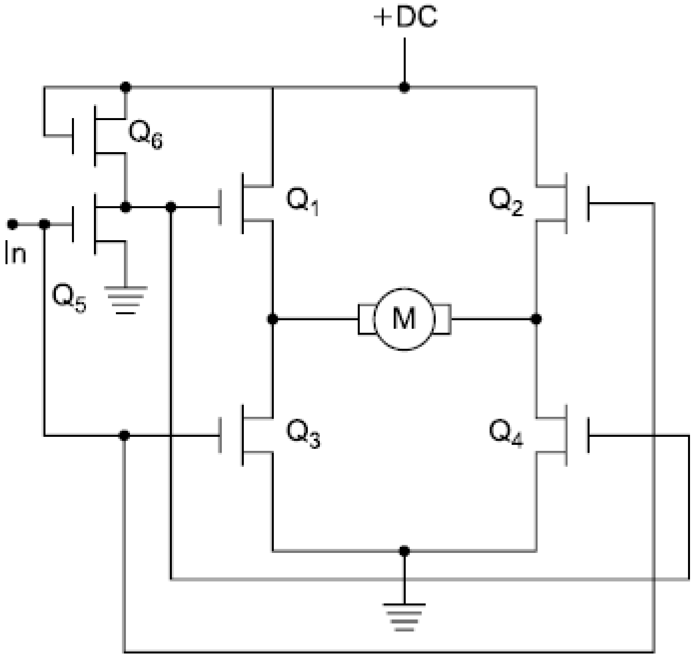


រូបភាព ២.៤៤ ៖ Solenoid

ម៉ូទ័រ

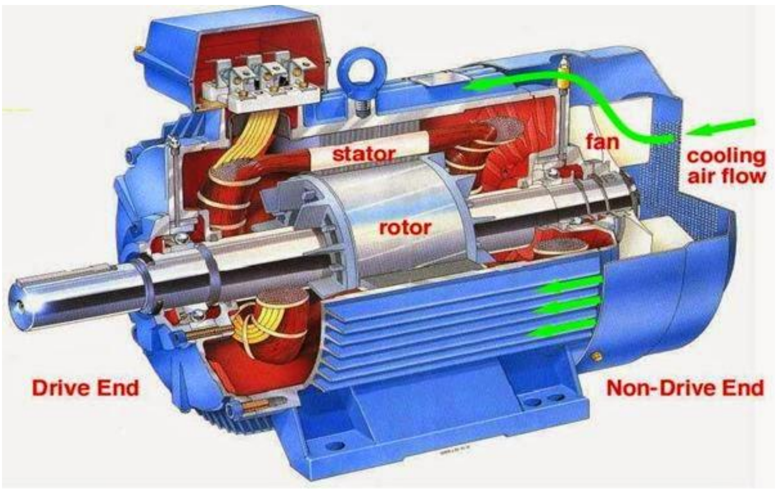
ម៉ូទ័រជាចលករយ៉ាងសំខាន់នៅក្នុងជីវភាពរស់នៅប្រចាំថ្ងៃ និងក្នុងឧស្សាហកម្ម ដែលបម្លែងថាមពលអគ្គីសនីទៅជាថាមពលម៉េកានិកក្នុងទម្រង់ជាចលនារង្វិល។ វាអាចជាម៉ូទ័រប្រើប្រាស់ចរន្តជាប់ ឬ ម៉ូទ័រ ប្រើប្រាស់ចរន្តឆ្លាស់។ ភាគច្រើនម៉ូទ័រតូចៗច្រើនប្រើប្រាស់ចរន្តជាប់ ចំណែកឯម៉ូទ័រធំៗជាម៉ូទ័រចរន្តឆ្លាស់ ដែលអាចប្រើប្រាស់ចរន្តអគ្គីសនី១ហ្វា ឬ ៣ហ្វា។

គេអាចផ្លាស់ប្តូរទិសដៅរង្វិលរបស់ម៉ូទ័រចរន្តជាប់ដោយគ្រាន់តែត្រលប់ប៉ូលទាំងពីរដែលភ្ជាប់ទៅនឹង ជនិតាអគ្គីសនី។ ការធ្វើបែបនេះមានការលំបាក ទើបគេប្រើ H-bridge ដូចនៅក្នុងរូបភាព ២.៤៥ ជំនួសវិញ។



រូបភាព ២.៤៥ ៖ H-bridge សម្រាប់ផ្លាស់ប្តូរទិសដៅវិលរបស់ម៉ូទ័រចរន្តជាប់

ចំណែកម៉ូទ័រចរន្តឆ្លាស់វិញ ប្រើប្រាស់ចរន្តឆ្លាស់មានហ្វ្រេកង់ 50 Hz ឬ 60 Hz អាស្រ័យទៅតាមប្រភពចរន្តអគ្គិសនីនៅតាមប្រទេសនីមួយៗ។ នៅកម្ពុជាយើង ចរន្តអគ្គិសនីប្រើប្រាស់ហ្វ្រេកង់ 50Hz។ ម៉ូទ័រចរន្តឆ្លាស់ដែលគេនិយមប្រើប្រាស់គឺម៉ូទ័រអាំងឌុចស្យុង (induction motor) ដូចបង្ហាញក្នុងរូបភាព ២.៤៦។ ល្បឿនជុំរបស់វាវិលអាស្រ័យនឹងចំនួនរបុំខ្សែ និងចំនួនប៉ូលរបស់ម៉ូទ័រ។ ដើម្បីប្តូរល្បឿនជុំត្រូវប្តូរហ្វ្រេកង់របស់ចរន្តឆ្លាស់។ ឧបករណ៍ប្តូរហ្វ្រេកង់ដែលគេនិយមប្រើប្រាស់ឈ្មោះថា Variable Frequency Drive (VFD)។



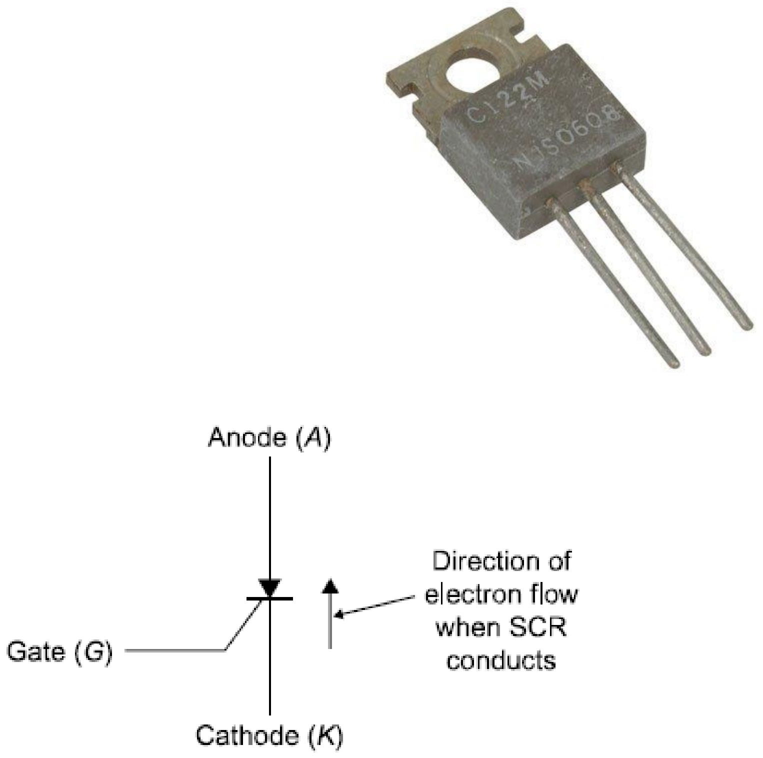
រូបភាព ២.៤៦ ៖ ម៉ូទ័រចរន្តឆ្លាស់ប្រភេទអាំងឌុចស្យុង

Thyristors

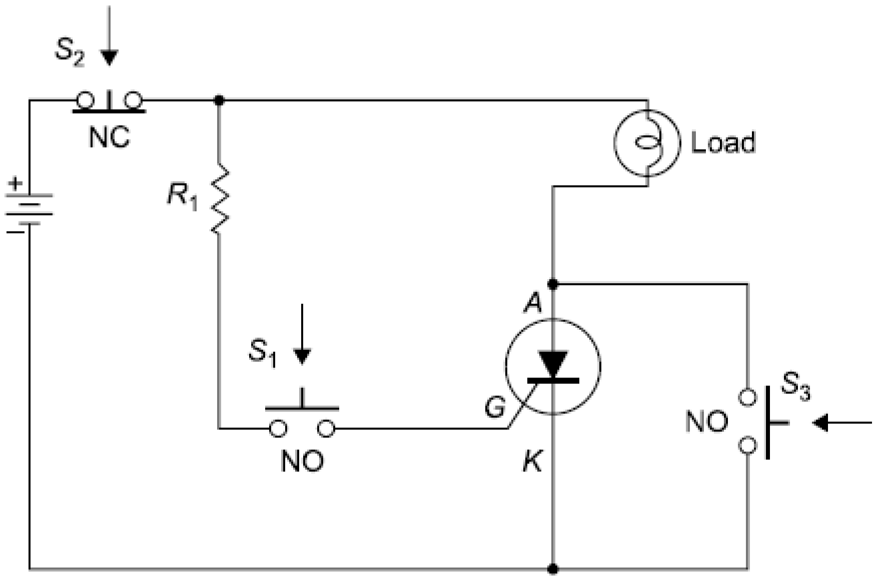
គេអាចប្រើប្រាស់ Thyristors ជំនួស Relays ដោយសារវាជាប្រភេទ solid-state ហេតុនេះល្បឿនបើកបិទកុងតាក់របស់វាលឿនជាង relay។ ប្រភេទ Thyristor សំខាន់ៗមាន Silicon-Controlled Rectifier (SCR), Triacs និង Diacs។

Silicon-Controlled Rectifier (SCR)

Silicon-Controlled Rectifier ដូចបង្ហាញក្នុងរូបភាព ២.៤៧ វាមានអាណូតតំណាងឲ្យអក្សរ A កាតូតតំណាងដោយអក្សរ K និង Gate តំណាងដោយអក្សរ G។ គេប្រើប្រាស់វាជាមួយចរន្តអគ្គីសនីមានថាមពលខ្ពស់ អាចមានតង់ស្យុងខ្ពស់ដែរ ដូចជា អំពូលភ្លើងអាចកែសម្រួលឲ្យភ្លឺខ្លាំង ឬ ខ្សោយ កែសម្រួល តង់ស្យុង និងបញ្ជាម៉ូទ័រ។



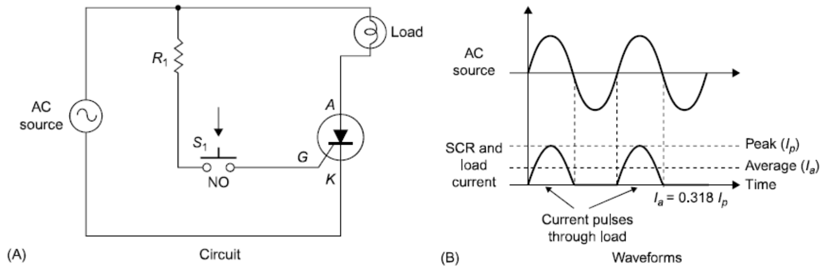
រូបភាព ២.៤៧ ៖ Silicon-controlled rectifier



រូបភាព ២.៤៨ ៖ ការប្រើប្រាស់ SCR ជាមួយចរន្តជាប់

ក្នុងរូបភាព ២.៤៨ អំពូលភ្លើងអាចបញ្ឆាបើកដោយចុចលើកុងតាក់ S1 រួចព្រលែងវិញ។ នៅពេលចុចកុងតាក់ S1 ចរន្តអគ្គីសនីផ្តល់ទៅឲ្យ SCR តាមរយៈរ៉េស៊ីស្តង់ R1 ទៅ gate ធ្វើឲ្យ SCR ចម្លងចរន្តអគ្គីសនី នាំឲ្យអំពូលភ្លើងភ្លឺ។ ដើម្បីបិទអំពូលភ្លើងវិញ គេអាចចុចកុងតាក់ S2 ឬ S3។

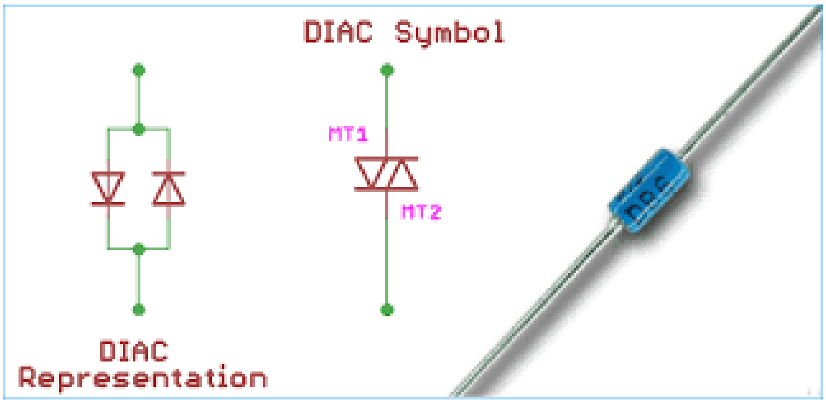
SCR ប្រើប្រាស់ជាមួយចរន្តឆ្លាស់ដូចបង្ហាញនៅក្នុងរូបភាព ២.៤៩។ ស្រដៀងទៅនឹងឌីយ៉ូតដែរ វាចម្លងចរន្តអគ្គីសនីតែមួយទិសទេ រូបភាព ២.៤៩ (B)។ រូបភាពខាងលើជាក្រាហ្វិកតំណាងឲ្យចរន្តឆ្លាស់ចេញពីប្រភព ចំណែកឯរូបភាពខាងក្រោម តំណាងឲ្យចរន្តដែលឆ្លងកាត់ SCR ដោយវាចម្លងតែមួយទិសប៉ុណ្ណោះ។



រូបភាព ២.៤៩ ៖ ការប្រើប្រាស់ SCR ជាមួយចរន្តឆ្លាស់

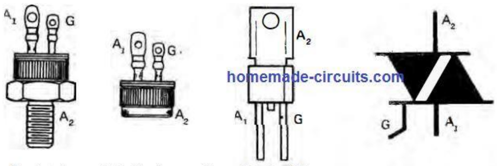
Diac

Diac ជាឧបករណ៍មានជើងពីរដូចបង្ហាញក្នុងរូបភាព ២.៥០ អាចចម្លងចរន្តទាំងទៅទាំងមកនៅពេលតង់ស្យុងលើសកម្រិតកំណត់របស់វាដែលមានតម្លៃពី 20 V ទៅ 45 V។ វាប្រៀបដូចជាឌីយ៉ូតពីរអញ្ចឹងដែរ។ គេប្រើប្រាស់ប្រាស់វាដើម្បីបើក ឬ បិទ SCR និង Triac។



រូបភាព ២.៥០ ៖ Diac

Triac



Construction and electrical connections of typical Triacs.

រូបភាព ២.៥១ ៖ រូបរាងនិងនិមិត្តសញ្ញារបស់ Triac

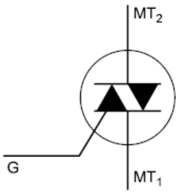


FIGURE 12.24 Schematic symbol for triac.

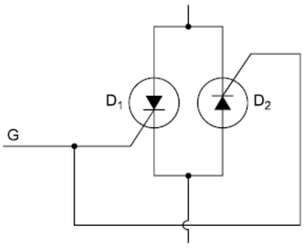
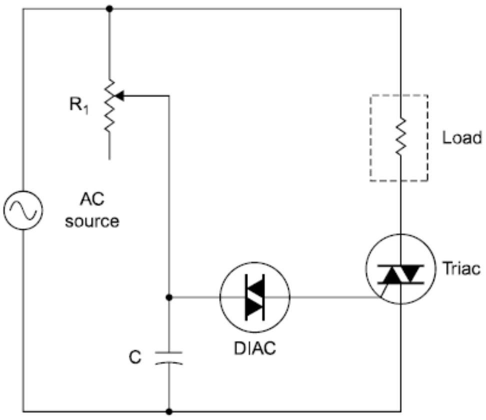


FIGURE 12.25 Equivalent circuit of triac shown with SCRs.

រូបភាព ២.៥២ ៖ Triac ស្រដៀងនឹង SCR ពីរ

រូបភាព ២.៥២ ជាសៀគ្វីសមមូលរបស់ Triac ដែលផ្គុំដោយ SCR ពីរ MT1 របស់ SCR ទី១ តភ្ជាប់ MT2 របស់ SCR ទី២ និង MT2 របស់ SCR ទី១ តភ្ជាប់ MT1 របស់ SCR ទី២ ហើយ gate របស់វាតភ្ជាប់គ្នា។

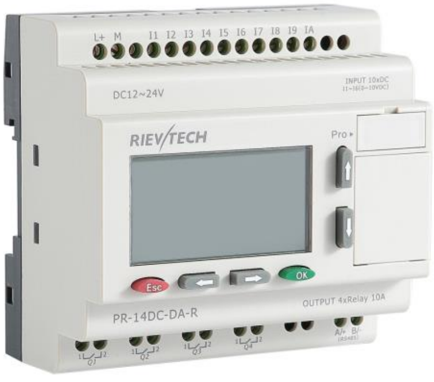
ក្នុងរូបភាព ២.៥៣ គេប្រើ Triac និង Diac ដើម្បីឲ្យភ្លើងភ្លឺខ្លាំង ឬ ខ្សោយ ដោយប្រើប្រាស់ប្រភពចរន្តឆ្លាស់។ សៀគ្វី RC តភ្ជាប់ diac ដើម្បីផ្តល់ចរន្តទៅឲ្យ gate របស់ triac ហើយវាចម្លងចរន្តខ្លាំង ឬ ខ្សោយ អាស្រ័យនឹងចរន្តដែលផ្តល់ទៅឲ្យ gate តាមរយៈ diac ដែលអាចកែសម្រួលបានតាមរយៈរ៉េស៊ីស្តង់ potentiometer R1។



រូបភាព ២.៥៣ ៖ ការប្រើប្រាស់ Triac ដើម្បីឲ្យភ្លើងភ្លឺខ្លាំង ឬ ខ្សោយ

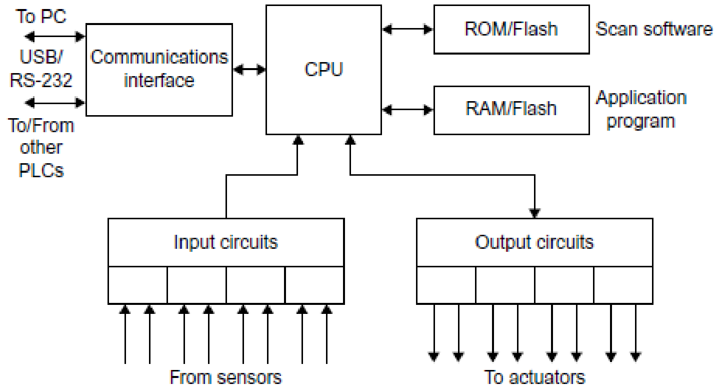
Programmable Logic Controllers (PLC)

Programmable Logic Controllers (PLC) ជាកុំព្យូទ័រប្រភេទពិសេសប្រើប្រាស់សម្រាប់បញ្ជាម៉ាស៊ីន ឬ ឧបករណ៍ក្នុងវិស័យឧស្សាហកម្ម ដោយស្វ័យប្រវត្តិ ដូចបង្ហាញក្នុងរូបភាព ២.៥៤។ វាខុសពីកុំព្យូទ័រធម្មតាដោយសារគេផលិតជាពិសេសសម្រាប់ប្រើប្រាស់ក្នុងឧស្សាហកម្មដែលមានលក្ខខណ្ឌក្តៅ រំញ័រខ្លាំង និងធូលីច្រើនជាដើម។ វាទទួលព័ត៌មានពីសេនស័រ ដើម្បីតាមដានដំណើរការ ហើយបញ្ចេញសញ្ញាដើម្បីបើកឧបករណ៍កម្តៅ ម៉ូទ័រ ស្នប់បូម Solenoids។ល។ អាស្រ័យទៅនឹងលក្ខខណ្ឌដែលគេកំណត់នៅក្នុងកម្មវិធីរបស់វា។



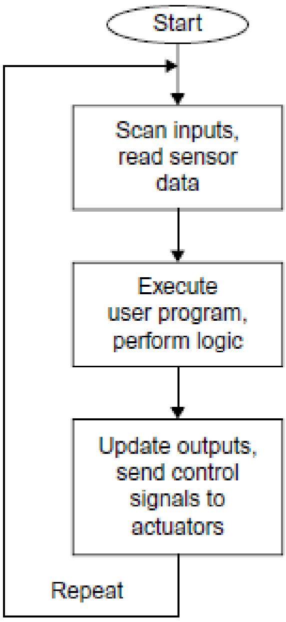
រូបភាព ២.៥៤ ៖ PLC

ប្លុកដ្យាក្រាមរបស់ PLC ដូចបង្ហាញក្នុងរូបភាព ២.៥៥ មាន CPU ប្រៀបដូចជាខួរក្បាលរបស់ PLC ដែលអាចដំណើរការទិន្នន័យ។ កម្មវិធីរបស់វារក្សាទុកនៅក្នុង ROM/flash ហើយត្រូវរក្សាទុកនៅក្នុង RAM/flash នៅពេលដំណើរការ ដើម្បីបញ្ជា CPU ឲ្យធ្វើការងារផ្សេងៗ វាក៏ជាកន្លែងរក្សាទុកទិន្នន័យបណ្ដោះអាសន្នដែលទទួលបានពីសេនស័រ តាមរយៈសៀគ្វីចូល (input circuit)។ CPU អាចបង្កើតបញ្ជាផ្សេងទៅ actuator តាមរយៈសៀគ្វីចេញ (output circuit) ឧទាហរណ៍ បើក ឬ បិទ ម៉ូទ័រ ជាដើម។ PLC អាចទំនាក់ទំនងជាមួយ PLC ផ្សេងទៀត ឬ កុំព្យូទ័រតាមរយៈ communication interface ដោយប្រើប្រាស់ខ្សែ USB, RS-232 ជាដើម។ ជាធម្មតា កម្មវិធីរបស់ PLC គេបង្កើតនៅក្នុងកុំព្យូទ័រ រួចគេអាចផ្ទេរវាចូលទៅក្នុង ROM/flash របស់ PLC តាមរយៈខ្សែ USB។



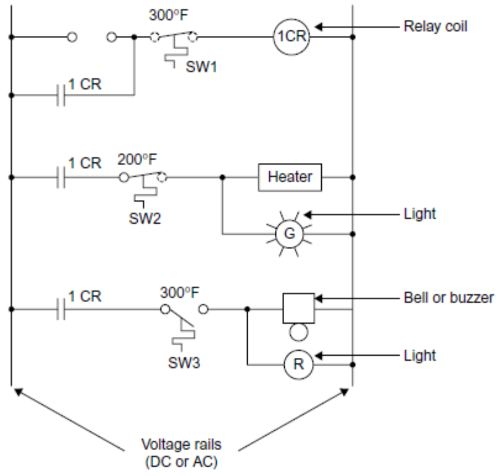
រូបភាព ២.៥៥ ៖ ប្លុកឌីយ៉ាក្រាមរបស់ PLC

វដ្តដំណើរការរបស់ PLC បង្ហាញនៅក្នុងរូបភាព ២.៥៦។ នៅពេល គេបើក PLC ដំបូង វាចាប់ផ្ដើមទទួលធាតុចូលនិងទិន្នន័យពីសេនស័រផ្សេងៗ អាស្រ័យទៅនឹងកម្មវិធីដែលយើងបានដាក់នៅក្នុង PLC វានឹងធ្វើបច្ចុប្បន្នភាពធាតុចេញ (output) រួចបង្កើតសញ្ញាបញ្ជាទៅ actuator។ វដ្តដំណើរការនេះ ធ្វើឡើងយ៉ាងលឿនរៀងរាល់ 1 ms ឬ យូរជាងនេះ អាស្រ័យតាមការបញ្ជារបស់យើងនៅក្នុងកម្មវិធីរបស់វា។



រូបភាព ២.៥៦ ៖ វដ្តដំណើរការរបស់ PLC

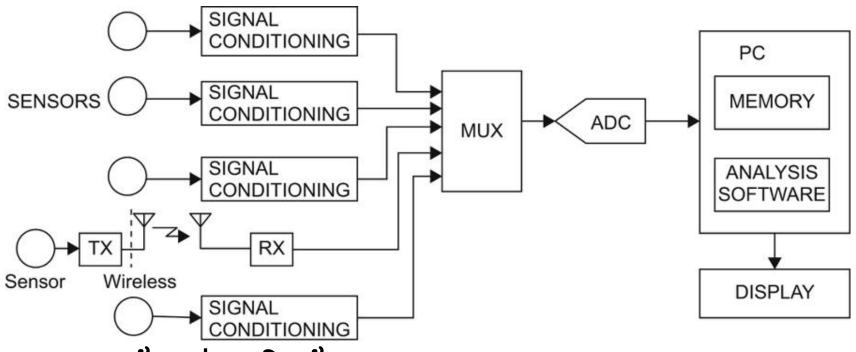
ការសរសេរកម្មវិធីសម្រាប់ PLC អាចប្រើប្រាស់ភាសាកុំព្យូទ័រផ្សេងៗ អាស្រ័យតាមក្រុមហ៊ុនផលិត។ គេអាចប្រើប្រាស់ភាសាកុំព្យូទ័រ BASIC ឬ C។ ជាធម្មតាភាសា BASIC មានភាពងាយស្រួលរៀនជាង C សម្រាប់អ្នកចាប់ផ្តើមរៀនដំបូង។ វិធីសរសេរកម្មវិធីមួយទៀតប្រើប្រាស់ក្រាហ្វិក (រូបភាពតំណាង) ហៅថា ladder logic ដូចរូបភាព ២.៥៧ ខាងក្រោម វាងដូចជណ្ដើរ។ បន្ទាត់ឈរសងខាងជាខ្សែចរន្ត អាចជាចរន្តឆ្លាស់ ឬជាចរន្តជាប់។ ចំណែកឯកាំនីមួយៗរបស់វា ជាលក្ខខណ្ឌដំណើរការនិងការតភ្ជាប់នៃគ្រឿងផ្គុំរបស់ប្រព័ន្ធ។ នៅក្នុងរូបភាព ២.៥៧ កាំទី១ មានកុងតាក់ដែលអាចបញ្ជាបើកបិទកុងតាក់របស់ coil relay 1CR ដែលតជាស៊េរីជាមួយ កុងតាក់សីតុណ្ហភាព SW1 ជាប្រភេទ NO ជាធម្មតាចម្លងចរន្តអគ្គិសនី ប៉ុន្តែវានឹងផ្តាច់ឈប់ឲ្យចរន្តអគ្គសនីឆ្លងទៅ relay CR1 នៅពេលសីតុណ្ហភាពធំជាង 300 °F (ប្រហែលជា 149 °C)។ កាំទី ២ មានកុងតាក់ 1CR តជាស៊េរីជាមួយកុងតាក់សីតុណ្ហភាព SW2 ដែលធម្មតាវាចម្លងចរន្តធ្វើឲ្យ heater ដំណើរការ និងអំពូលពណ៌បៃតងដែលតជាខ្នែងជាមួយវាភ្លឺដែរ។ រហូតសីតុណ្ហភាពឡើងដល់ 200 °F (ប្រហែលជា 93 °C) វានឹងផ្តាច់ឈប់ឲ្យចរន្តឆ្លងកាត់។ ប្រសិនបើសីតុណ្ហភាពឡើងលើស 93 °C ដោយសារ SW2 ខូច SW1 នឹងផ្តាច់ចរន្តទៅ 1CR បើកសៀគ្វីនៅកាំទី២ ធ្វើឲ្យ heater និងអំពូលពណ៌បៃតងឈប់ដំណើរការ និងបិទសៀគ្វីនៅកាំទី៣ ធ្វើឲ្យមានសំលេងព្រមាននិងអំពូលពណ៌ក្រហមភ្លឺ។ ទី៣ មានកុងតាក់ 1CR តជាស៊េរីជាមួយកុងតាក់សីតុណ្ហភាព SW3 ដែលធម្មតាវាមិនចម្លងចរន្តធ្វើឲ្យ Bell/Buzzer បញ្ចេញសំលេងព្រមាន ឬភ្លឺអំពូលពណ៌ក្រហមដែលតជាខ្នែងជាមួយវាទេ។ រហូតសីតុណ្ហភាពឡើងដល់ 300 °F (ប្រហែលជា 149 °C) កុងតាក់ 1CR នឹងភ្ជាប់ឲ្យចរន្តឆ្លងកាត់ធ្វើឲ្យលឺសម្លេងព្រមាន និងអំពូលពណ៌ក្រហមភ្លឺ។



រូបភាព ២.៥៧ ៖ Ladder logic សម្រាប់បញ្ជាប្រព័ន្ធកម្តៅ

២.៥ ប្រព័ន្ធកត់ត្រាទិន្នន័យ (Data Acquisition system)

ប្រព័ន្ធកត់ត្រាទិន្នន័យ Data Acquisition system សរសេរកាត់ DAQ ដូចមានបង្ហាញក្នុង រូបភាព ២.៥៨ ប្រើប្រាស់សម្រាប់កត់ត្រាទិន្នន័យពីសេនស័រច្រើនប្រភេទនៅពេលធ្វើតេស្ត ឬ តាមដានដំណើរការរបស់ម៉ាស៊ីន ឬ ខ្សែច្រវាក់ផលិតកម្ម ដើម្បីដឹងថាវាដំណើរធម្មតា ឬ មានបញ្ហា។ ប៉ារ៉ាម៉ែត្រដែលគេឧស្សាហ៍វាស់វែង និងតាមដានមាន សីតុណ្ហភាពសំណើម សំពាធ កំលាំង ស្រ្តេស ទីតាំង និងលំហូរ។ល។ ជាតុសំខាន់ៗរបស់វាមាន សេនស័រ សៀគ្វីលក្ខខណ្ឌសញ្ញា multiplexer, ADC និងកុំព្យូទ័រ។ សញ្ញាពីសេនស័រអាចបញ្ជូនតាមខ្សែ ឬអត់ខ្សែ (WIFI) ត្រូវបានឆ្លងកាត់សៀគ្វីលក្ខខណ្ឌសញ្ញាដើម្បីបង្កើនថាមពល ច្រោះសញ្ញារំខានចេញ ប្តូរកម្រិត ឬ ការកែសម្រួលផ្សេងៗទៀត។ សញ្ញាពីសៀគ្វីលក្ខខណ្ឌសញ្ញាត្រូវបានជ្រើសរើសដើម្បីបំលែងទៅជាឌីជីថលដោយ multiplexer រួចបំលែងជាឌីជីថលដោយ Analog to Digital Converter (ADC) ដើម្បីអាចរក្សាទុក និងប្រើប្រាស់ដោយកុំព្យូទ័រ។ នៅក្នុងកុំព្យូទ័រ ទិន្នន័យរក្សាទុកនៅក្នុងអង្គចង់ចាំ (memory) នឹងវិភាគដោយកម្មវិធីផ្សេងៗ រួចបង្ហាញលទ្ធផលនៅលើអេក្រង់។



រូបភាព ២.៥៨ ៖ ប្រព័ន្ធកត់ត្រាទិន្នន័យ

ជំពូកទី៣
ម៉ាយក្រូកុនត្រូល័រ (Microcontroller) និងការប្រើប្រាស់ក្នុងការកត់ត្រាទិន្នន័យនិងស្វ័យប្រវត្តិកម្ម

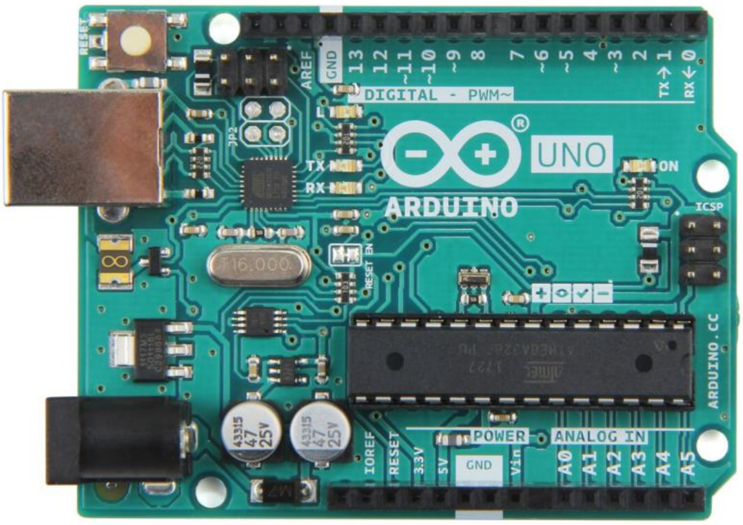
៣.១ ការណែនាំពីម៉ាយក្រូកុនត្រូល័រ Arduino

ម៉ាយក្រូកុនត្រូល័រមានប្រើប្រាស់ច្រើននៅក្នុងការរស់នៅប្រចាំថ្ងៃ និងក្នុងឧស្សាហកម្ម។ ឧបករណ៍ប្រើប្រាស់ក្នុងផ្ទះដូចជា ទូរទឹកកក ទូរទស្សន៍ឆ្លាតវៃ ម៉ាស៊ីនបោកខោអាវ ឧបករណ៍បញ្ជាពីចម្ងាយ ច្រើនតែមានប្រើប្រាស់ម៉ាយក្រូកុនត្រូល័រនៅក្នុងនោះ។ នៅក្នុងវិស័យកសិកម្ម យើងអាចប្រើប្រាស់វាសម្រាប់ធ្វើស្វ័យប្រវត្តិកម្មក្នុងដំណើរការដាំដុះ ចិញ្ចឹមសត្វ វារីវប្បកម្ម តាមដាននិងកត់ត្រាទិន្នន័យដំណើរការផ្សេងៗ ឧទាហរណ៍ ប្រព័ន្ធត្រួតពិនិត្យនិងបញ្ជាក្នុងផ្ទះសំណាញ់ រោងចិញ្ចឹមសត្វ និងការធ្វើតេស្តម៉ាស៊ីននិងឧបករណ៍កសិកម្ម ជាដើម។ Arduino គឺជាម៉ាយក្រូកុនត្រូល័រដែលពេញនិយម ដោយសារងាយស្រួលប្រើប្រាស់ រកទិញ និងមានអ្នកបង្កើតមុខងារច្រើន និងជា open source។

៣.១.១ ប្រភេទផ្សេងៗនៃ Arduino

ប្រភេទ Arduino សំខាន់ៗមាន Uno, Nano, Mega ដែលបានផលិតឡើងដើម្បីសម្របតាមតម្រូវការប្រើប្រាស់ប្រភេទផ្សេងៗគ្នា។ ការប្រើប្រាស់ខ្លះត្រូវការចំនួន input pin និង output pin ច្រើន កម្មវិធីមានភាពស្មុគស្មាញ និងការប្រើប្រាស់ខ្លះសាមញ្ញ ត្រូវការទំហំតូច និងតម្លៃទាប។

Arduino Uno



រូបភាព ៣.១ ៖ Board របស់ Arduino Uno

ក្នុងរូបភាព ៣.១ ជា board របស់ Arduino Uno គ្រឿងផ្សំសំខាន់ៗរបស់វាមានបង្ហាញក្នុងរូបភាព ៣.២។

រូបភាព ៣.២ ៖ គ្រឿងផ្សំសំខាន់ៗរបស់ Arduino Uno

Reset Switch ៖ ប្រើប្រាស់សម្រាប់បើក Arduino Uno ឡើងវិញ ក្រោយពេល upload កម្មវិធី (Sketch) របស់វាចូលរួចហើយ ដើម្បីឲ្យវាចាប់ផ្តើមដំណើរការ។

USB connector ៖ ប្រើប្រាស់សម្រាប់ផ្តល់ថាមពលអគ្គីសនីទៅឲ្យ Arduino Uno ក៏ដូចជាទំនាក់ទំនងរវាងវាជាមួយកុំព្យូទ័រដូចជា upload sketch ពីកុំព្យូទ័រចូលទៅក្នុង Arduino Uno ឬសម្រាប់ផ្ញើ ឬ ទទួលព័ត៌មានពីកុំព្យូទ័រ (Serial Monitor)។

USB interface chip ៖ មានមុខងារជួយឲ្យ Arduino អាចប្រើប្រាស់ USB connector បាន។

Crystal Oscillator ៖ ជាអ្នកបង្កើត clock signal ដែលម៉ាយក្រូកុនត្រូល័រត្រូវការសម្រាប់ដំណើរការ។

Voltage Regulator ៖ ជាអ្នករក្សាលំនឹងតង់ស្យុងអគ្គីសនីឲ្យមានស្ថេរភាព ឧទាហរណ៍ 5V ឬ 3.3 V សម្រាប់ផ្តល់ថាមពលដល់សេនស័រ ឬ actuator ដែលតភ្ជាប់ទៅនឹង Arduino។ វាទទួលបានថាមពលអគ្គិសនី ពី USB Connector, Power Port ឬ Vin នៅលើ board។

Power Port ៖ សម្រាប់ដោតដុំភ្លើង ផ្តល់ភ្លើងដល់ Ardunio តង់ស្យុងដែលអាចប្រើប្រាស់បានមានពី 7V-12V។

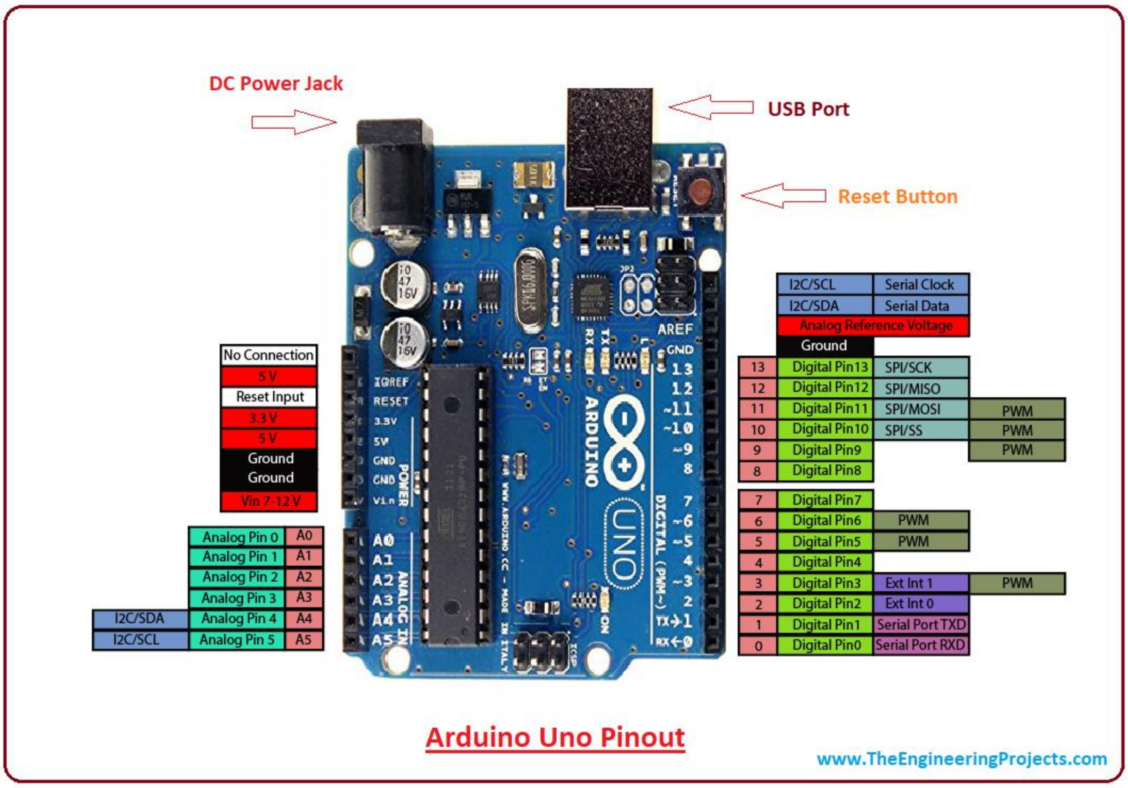
Analog input pins ៖ សម្រាប់តភ្ជាប់សេនស័រប្រភេទ analog ទៅនឹង Arduino Uno។

Microcontroller ៖ ជា IC របស់មីក្រូកុនត្រូល័រ។

TX RX LED ៖ ជាឌីយ៉ូតពន្លឺដែលនឹងភ្លឺពេល Arduino ផ្ញើ ឬ ទទួលព័ត៌មានតាម serial communication ដូចជារវាង Arduino Uno និងកុំព្យូទ័រជាដើម។

Digital pin ៖ សម្រាប់តភ្ជាប់សេនស័រ ឬ actuator ប្រភេទឌីជីថល។

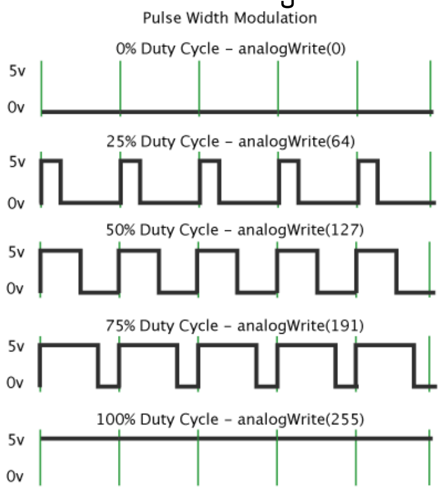
ព័ត៌មានពី pin របស់ Arduino Uno មានបង្ហាញនៅក្នុងរូបភាព ៣.៣។



រូបភាព ៣.៣៖ Pin របស់ Arduino Uno

ប្រភព ៖ https://www.theengineeringprojects.com

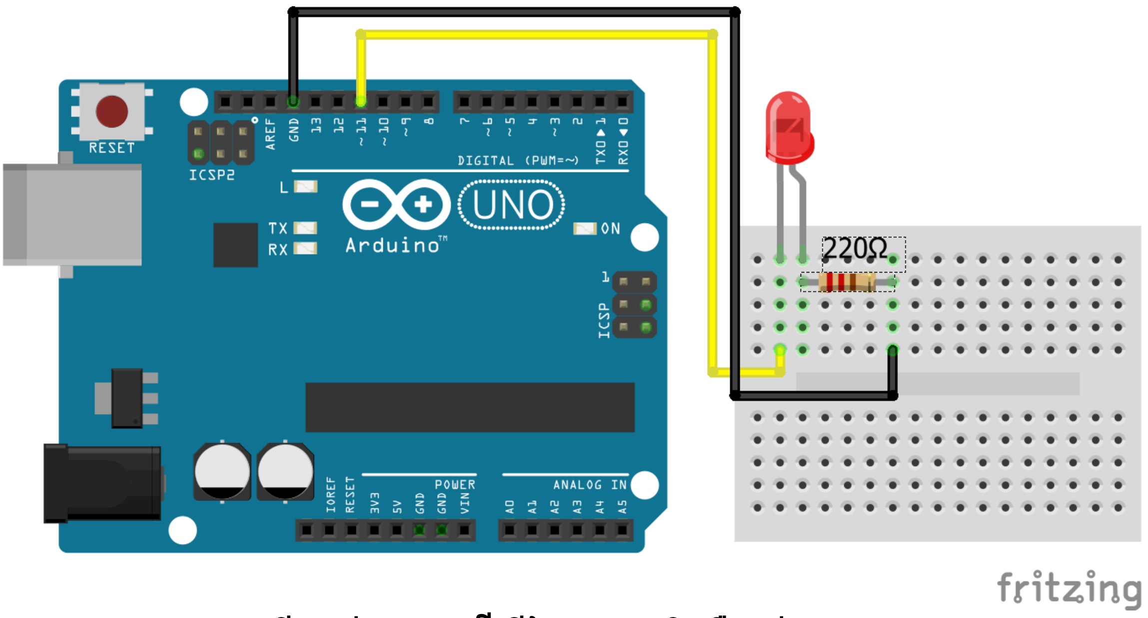
Pin analog ឬ ឌីជីថលខ្លះមានមុខងារច្រើន ដូចជា Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI) ឬ Pulse Width Modulation (PWM)។ ឧទាហរណ៍ analog pin A4, A5 អាចប្រើប្រាស់តភ្ជាប់ប្រភេទឧបករណ៍ I2C ចំណែកឯ digital pin D10, D11, D12, D13 ប្រើប្រាស់សម្រាប់តភ្ជាប់ជាមួយឧបករណ៍ប្រភេទ SPI។ Digital pin D3, D5, D6, D9, D10, D11 អាចប្រើប្រាស់ជា PWM ជាចរន្តឆ្លាស់មានសញ្ញារាងជាចតុកោណកែងដែលអាចប្រែប្រួលប្រេកង់ និង duty cycle (មើលរូបភាព ៣.៤) របស់វា ធ្វើឲ្យ LED ភ្លឺខ្លាំងឬខ្សោយ ឬម៉ូទ័រចរន្តជាប់វិលល្បឿនជុំលឿន ឬ យឺត។



រូបភាព ៣.៤ ៖ PWM ដែលមាន duty cycle ផ្សេងៗគ្នា

ឧទាហរណ៍ ក្នុងរូបភាព ៣.៥ LED ត្រូវបានតភ្ជាប់ទៅនឹង Arduino Uno ដោយកាតូតរបស់វាតភ្ជាប់ទៅនឹង GND (ម៉ាស់) របស់ Ardunio Uno។ ចំណែកឯអាណូតរបស់វាត្រូវបានតភ្ជាប់ជាស៊េរីជាមួយរេស៊ីស្តង់ 220 Ω មុននឹងតភ្ជាប់ទៅ digital pin D11 ដែលជាប្រភេទ PWM។ រេស៊ីស្តង់ដែលតជាស៊េរីជាមួយ LED ជួយបញ្ចុះតង់ស្យុងរបស់ D11 ដែលជាធម្មតា 5V ប្រសិនបើតផ្ទាល់ទៅអាណូតរបស់ LED អាចធ្វើឲ្យវាឆេះខូច។

នៅក្នុងរូបភាព ៣.៦ ជាកម្មវិធី (sketch) ដែលប្រើប្រាស់ក្នុងឧទាហរណ៍នេះ។



រូបភាព ៣.៥៖ ការប្រើប្រាស់ PWM ដើម្បីប្រែប្រួលកម្រិតភ្លឺរបស់ LED

Code

```
int LEDpin = 11; // define the pin of the LED
int bright = 0; // initial value of LED brightness
int incremt = 5; // incremental change in PWM frequency
int time = 100; // time period the PWM frequency is changing
void setup( )
{
  pinMode( LEDpin, OUTPUT ); // define the LEDpin as output pin
}
void loop( )
{
  analogWrite( LEDpin, bright ); // set LED brightness as PWM signal
  delay( time ); // wait for a time period
  bright = bright + incremt; // increment LED brightness
  // if the brightness is out of range, reduce brightness
  if ( bright <=0 || bright >=255 ) incremt = - incremt;
}
```

រូបភាព ៣.៦៖ Sketch សម្រាប់ធ្វើឲ្យ LED ប្រែប្រួលកម្រិតភ្លឺ

Arduino Sketch ប្រើប្រាស់ភាសាសរសេរកម្មវិធី C។ នៅទីនេះយើងនឹងមិននិយាយពន្យល់លម្អិតពីមុខងាររបស់ Sketch តាមជួរនីមួយៗទេ ទុកសម្រាប់ពេលក្រោយ ដោយសារគ្រាន់តែចង់ឲ្យអ្នកអានយល់ពី PWM ប៉ុណ្ណោះ។ នៅពេលដំណើរការ LED នឹងភ្លឺខ្លាំងឡើងៗ រួចហើយភ្លឺខ្សោយទៅៗវិញ វដ្តនេះនឹងផ្លាស់ប្តូរចុះឡើង ដោយសារកូដ

analogWrite(LEDpin, bright);

បញ្ចេញសញ្ញា PWM ទៅ pin D11 (ក្នុងកូដតាងដោយអថេរ LEDpin) ហើយ duty cycle របស់វាតាងដោយអថេរ bright ដែលតម្លៃរបស់វាប្រែប្រួលនៅចន្លោះ 0 (មិនភ្លឺ) ទៅ 255 (ភ្លឺខ្លាំងបំផុត) នៅក្នុងកូដ

bright = bright + incremt; // increment LED brightness

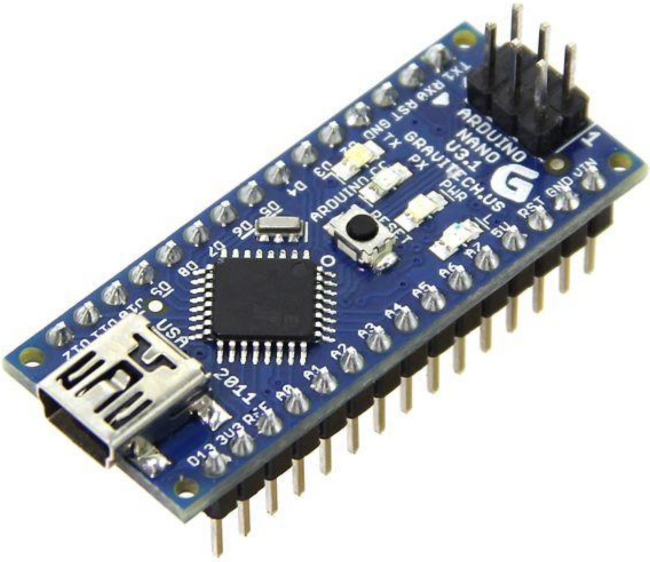
ដោយបូកបន្ថែម 5 (incremt) ក្នុង១វដ្តដំណើរការរបស់ Arduino

តម្លៃអថេរ incremt ត្រូវបានប្តូរសញ្ញា (ពីវិជ្ជមានទៅអវិជ្ជមាន ឬ ពីអវិជ្ជមានទៅវិជ្ជមាន) នៅពេលតម្លៃអថេរ bright តូចជាង ឬ ស្មើ 0 ឬ ធំជាង ឬ ស្មើ 255 ដោយកូដ

if (bright <=0 || bright >=255) incremt = - incremt;

Arduino Nano

Arduino Nano ស្រដៀង Arduino Uno ដែរ ប៉ុន្តែវាមានទំហំតូចជាង ដូចបង្ហាញក្នុងរូបភាព ៣.៧។ ប៉ុន្តែអត់មានកន្លែងដោតដុំភ្លើង និងប្រើប្រាស់ Mini-B USB។



រូបភាព ៣.៧ ៖ Arduino Nano

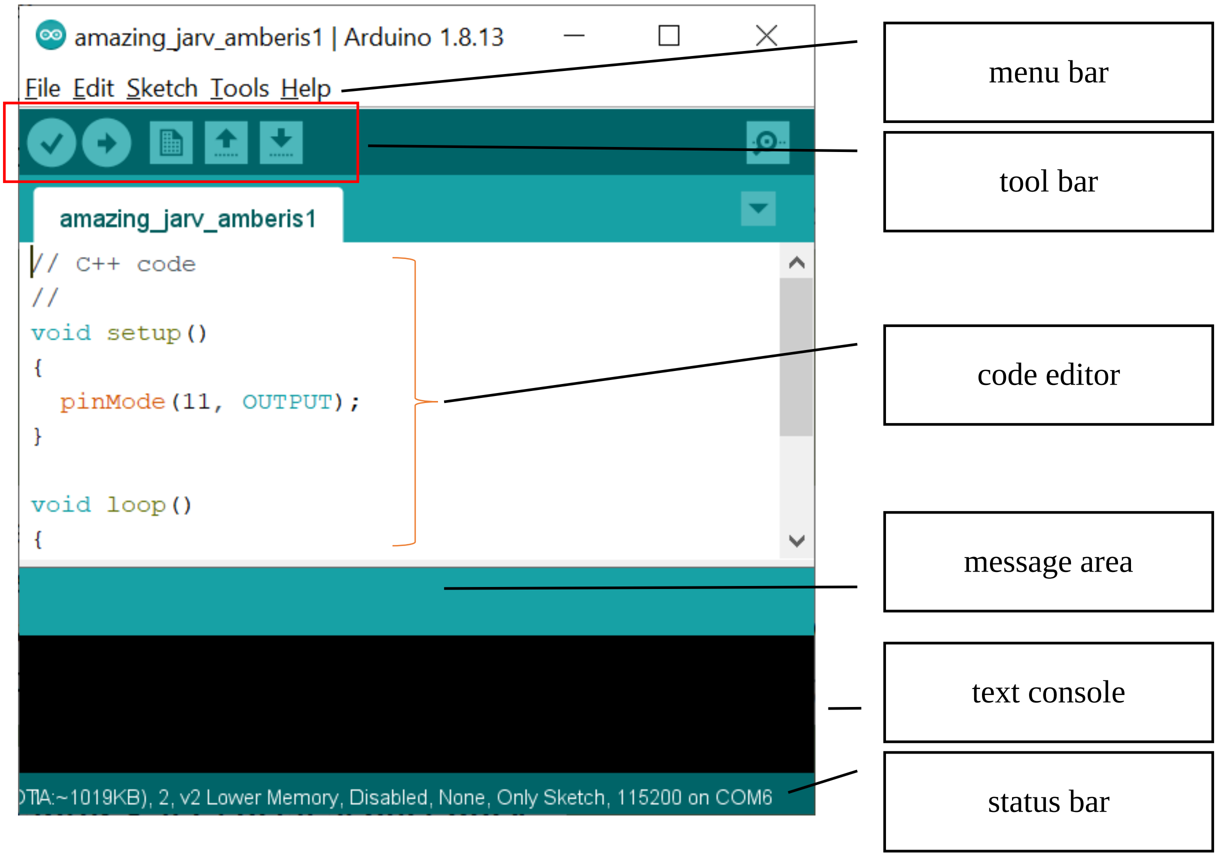
៣.១.២ សេនស័ររបស់ Arduino

មានសេនស័រច្រើនប្រភេទដែលអាចប្រើប្រាស់ជាមួយ Arduino បានដូចបង្ហាញក្នុងរូបភាព ៣.៨។ ឧទាហរណ៍ សេនស័រវាស់សីតុណ្ហភាពនិងសំណើមបរិយាកាសខ្យល់ប្រភេទ DHT11 និង DHT22 ជាដើម។

រូបភាព ៣.៨ ៖ ប្រភេទសេនស័រសំខាន់ៗដែលអាចប្រើប្រាស់ជាមួយ Arduino

៣.១.៣ Arduino IDE

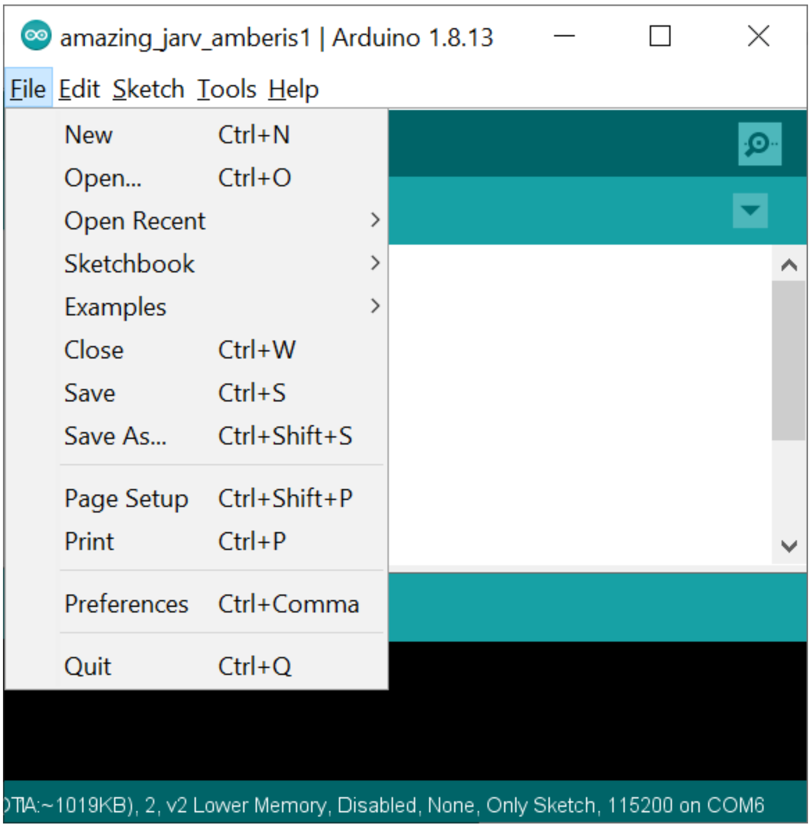
Arduino IDE ជាកម្មវិធីកុំព្យូទ័រប្រភេទ Integrated Development Environment (IDE) ដែលអាចប្រើប្រាស់សម្រាប់សរសេរកម្មវិធី (Sketch) ប្រើប្រាស់ជាមួយ Arduino ដូចបង្ហាញនៅក្នុងរូបភាព ៣.៩។ គេអាចប្រើប្រាស់វាសម្រាប់ កែសម្រួល sketch ផ្ទៀងផ្ទាត់ និង upload ទៅក្នុង Arduino។ វាក៏មាន Serial Monitor ដែលអាចឲ្យយើងបង្ហាញព័ត៌មានពីប្រតិបត្តិការ និងបញ្ជូនព័ត៌មាន ឬ បញ្ជា ទៅ Arduino។ ដូចកម្មវិធីភាគច្រើនដែរ នៅក្នុង window របស់វាមាន menu bar, toolbar, code editor (សម្រាប់កែសម្រួល sketch), message area, text console និង status bar។



រូបភាព ៣.៩ ៖ ផ្នែកផ្សេងៗនៃ window របស់ Arduino IDE

Menu bar មាន File, Edit, Sketch, Tools និង Help។ ខាងក្រោមនេះជាការពន្យល់ដោយសង្ខេប menu ទាំងនេះ។

File Menu

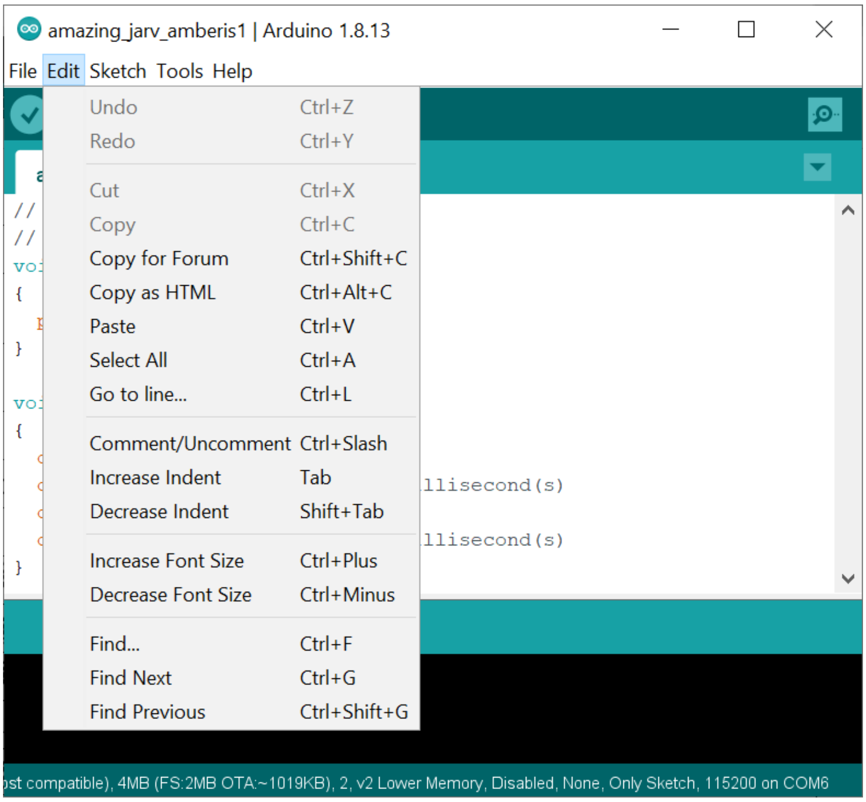


រូបភាព ៣.១០ ៖ File menu

File menu ដូចបង្ហាញក្នុងរូបភាព ៣.១០ អាចប្រើប្រាស់សម្រាប់ជ្រើសរើសមុខងារពាក់ព័ន្ធនឹង file ។ New សម្រាប់បង្កើត sketch ថ្មី។ Open សម្រាប់បើក sketch ដែលមានស្រាប់នៅក្នុងកុំព្យូទ័រ។ Open Recent សម្រាប់បើកឯកសារដែលទើបប្រើប្រាស់ថ្មីៗ។ Examples ជាឧទាហរណ៍ sketch ដែលភ្ជាប់មកជាមួយ Arduino IDE, library ឬ board វាបង្ហាញពីរបៀបប្រើប្រាស់សំខាន់ៗ ដែលយើងអាចយកមកកែសម្រួលបន្ថែមតម្រូវទៅនឹងការប្រើប្រាស់ជាក់ស្តែង។ Close ប្រើប្រាស់សម្រាប់បិទ sketch។ Save សម្រាប់ថតទុក sketch ទៅក្នុងកុំព្យូទ័រ ហើយអាចបើកមកវិញនៅពេលក្រោយ។ Save As សម្រាប់ថតទុក sketch ដោយដាក់ឈ្មោះថ្មីឲ្យវា។ Page Setup សម្រាប់កែសម្រួលក្រដាសនៅពេលចង់ព្រីន sketch ចេញ។ Print សម្រាប់បោះពុម្ព sketch ប្រើប្រាស់ព្រីនធ័រ។ Preferences អាចប្រើប្រាស់ដើម្បីកែសម្រួលមុខងាររបស់ Arduino IDE ជាពិសេសប្រើប្រាស់សម្រាប់បន្ថែម board ថ្មីៗ ទៅក្នុង Arduino IDE។ Quit សម្រាប់ចេញពីកម្មវិធី Arduino IDE។

Edit Menu

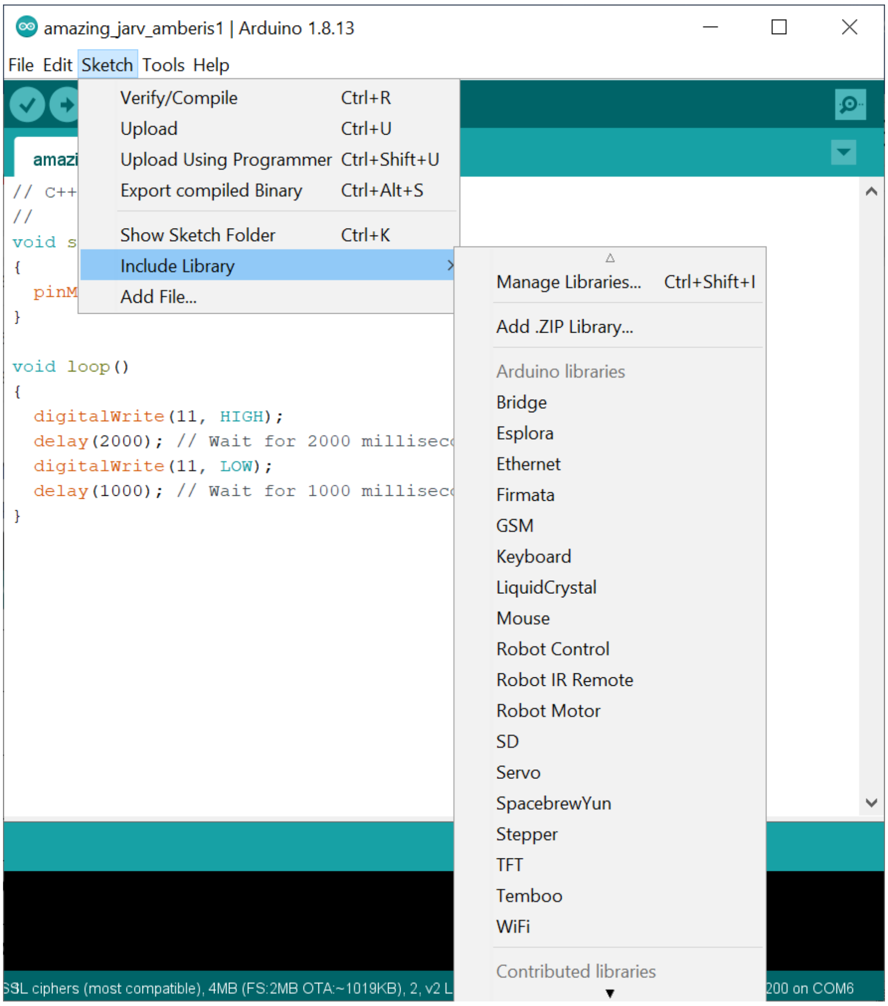
គេប្រើប្រាស់ Edit menu ដូចបង្ហាញក្នុងរូបភាព ៣.១១ ដើម្បីកែសម្រួល sketch។



រូបភាព ៣.១១ ៖ Edit menu

Undo សម្រាប់ត្រលប់ទៅមុនពេលកែ sketch ចំណែកឯ Redo សម្រាប់ត្រលប់ទៅមុនពេល Undo ។ Cut សម្រាប់កាត់ផ្នែកនៃ sketch ដែលបានជ្រើសរើស។ Copy សម្រាប់ថតចម្លង ផ្នែកនៃ sketch ដែលបានជ្រើសរើស។ Copy for Forum សម្រាប់ថតចម្លង ផ្នែកនៃ sketch ដែលបានជ្រើសរើស ដើម្បីយកទៅ Paste នៅក្នុង Forum របស់ Arduino។ Copy as HTML សម្រាប់ថតចម្លង ផ្នែកនៃ sketch ដែលបានជ្រើសរើស ជាទម្រង់ HTML tag ដែលអាចប្រើប្រាស់ជា web page។ Paste សម្រាប់ដាក់មកក្នុង code editor វិញ អត្ថបទដែលបាន Cut ឬ Copy ពី Arduino IDE ឬ ពីកម្មវិធីផ្សេង។ Select All ជ្រើសរើស sketch ទាំងអស់។ Go to line ប្រាប់ឲ្យ cursor រត់ទៅជួរណាមួយនៅក្នុង code editor។ Comment/Uncomment សម្រាប់ដាក់ ឬ ដក comment នៅក្នុង sketch ដែលបានជ្រើសរើសជាមុន ឬ ជួរដែល cursor កំពុងបិតនៅ។ Increase Indent បន្ថែម Tab ចំណែកឯ Decrease Indent ដក Tab វិញ។ Increase Font Size បង្កើនទំហំ font នៅក្នុង code editor ចំណែកឯ Decrease Font Size បន្ថយទំហំ font វិញ។ Find សម្រាប់រកពាក្យ ឬ កថាខណ្ឌនៅក្នុង sketch។ Find Next រកពាក្យបន្ទាប់ដែលស្វែងរកដោយ Find។ Find Previous រកពាក្យមុនដែលស្វែងរកដោយ Find.

Sketch Menu



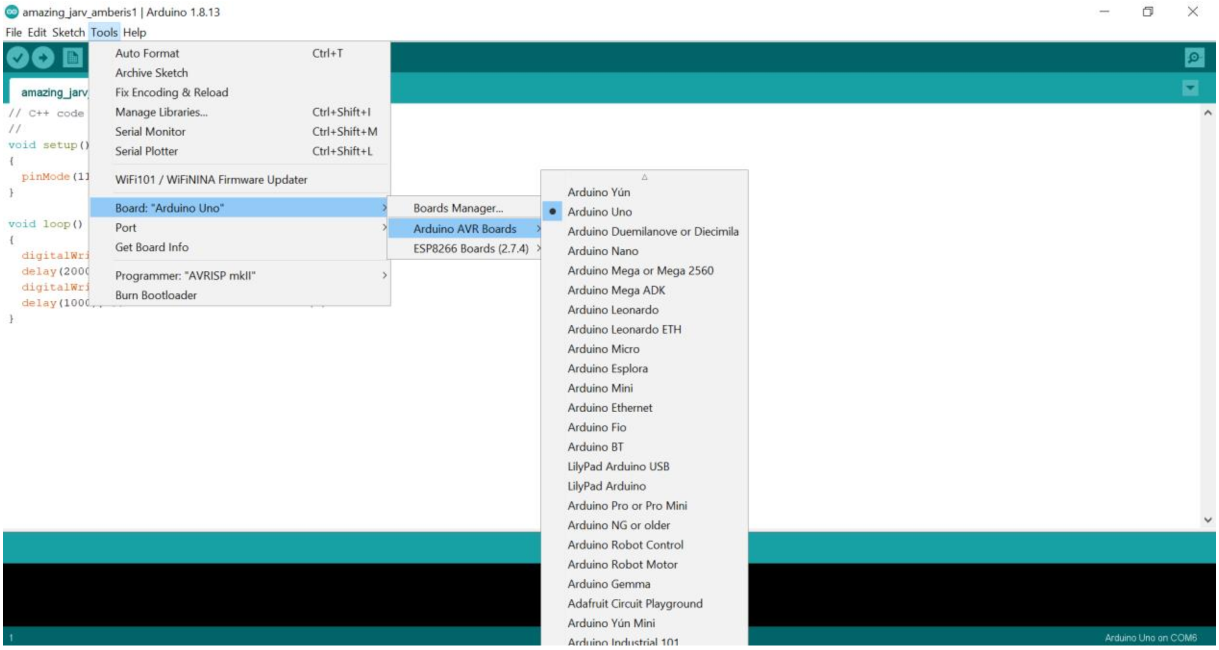
រូបភាព ៣.១២ ៖ Sketch menu

Sketch menu ដូចបង្ហាញក្នុងរូបភាព ៣.១២ ប្រើប្រាស់សម្រាប់ធ្វើការជាមួយ sketch។ Verify/Compile សម្រាប់ផ្ទៀងផ្ទាត់ ឬបំលែង sketch ទៅជាកូដ binary ដែល Arduino អាចដំណើរការបាន។ Upload គឺសម្រាប់ផ្ទេរ កម្មវិធីពី Arduino IDE ទៅក្នុង Arduino board។ Upload Using Programmer សម្រាប់ upload sketch ដោយប្រើប្រាស់ Programmer (ជា board ដែលអាចដោត chip របស់ម៉ាយក្រូកុនត្រូល័រ)។ Export compiled Binary សម្រាប់នាំចេញកូដដែលបម្លែងរួចហើយ។ Show Sketch Folder សម្រាប់បើក file explorer ហើយបង្ហាញ folder ដែលរក្សាទុក sketch របស់ Arduino។ Include Library សម្រាប់បន្ថែម library ផ្សេងៗរបស់ Arduino ទៅក្នុង sketch របស់ Arduino ដោយវានឹងបន្ថែម ជួរដែលមាន

#include <library.h>

Add File សម្រាប់បន្ថែម sketch ទៅក្នុង Arduino IDE ដោយវានឹងបន្ថែមជា file ថ្មី។

Tools Menu

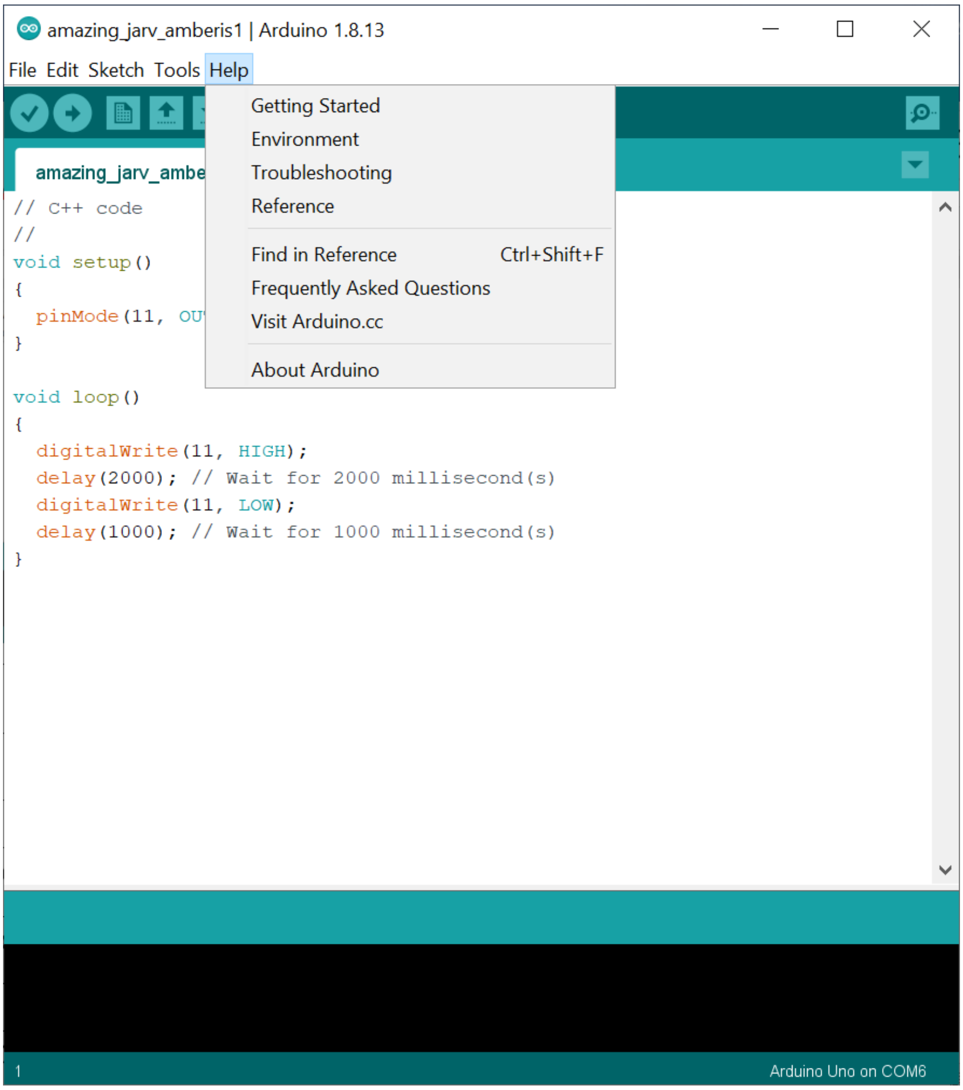


រូបភាព ៣.១៣ ៖ Tool menu

Auto Format សម្រាប់រៀបចំ sketch នៅក្នុង code editor ដើម្បីឲ្យមានសណ្តាប់ធ្នាប់ល្អ។ Archive Sketch សម្រាប់រក្សាទុក sketch ជា zip file ហើយរក្សាទុក folder ដូច sketch ដែរ។ Fix Encoding & Reload ប្រើប្រាស់សម្រាប់ជួសជុលបញ្ហា encoding របស់ sketch ពី editor ឬ operating system ផ្សេងៗ។ Manage Libraries អាចប្រើប្រាស់ដើម្បី install ឬ update library របស់ Arduino ដើម្បីអាចប្រើប្រាស់ជាមួយ module ផ្សេងៗដើម្បីបង្កើនមុខងាររបស់ Arduino។ Serial Monitor សម្រាប់បង្ហាញ Serial Monitor ដែលអាចបង្ហាញព័ត៌មាន និងទំនាក់ទំនងជាមួយ Arduino។ Serial Plotter ដើម្បីបង្ហាញទិន្នន័យទទួលបានពី Arduino ជាក្រាហ្វិក។ Board ប្រើប្រាស់ដើម្បីជ្រើសរើសប្រភេទ board ដែលតភ្ជាប់ទៅនឹង Arduino IDE។ Port សម្រាប់ជ្រើសរើសប្រភេទ port ដែល Arduino តភ្ជាប់ទៅកុំព្យូទ័រ។ Get Board Info ដើម្បីបង្ហាញព័ត៌មានពី board របស់ Arduino។ Programmer ជ្រើសរើស hardware programmer ដើម្បីផ្ទេរកម្មវិធីទៅ board ឬ chip មិនប្រើប្រាស់ខ្សែ USB ទេ។ ជាធម្មតាគេមិនត្រូវការប្រើប្រាស់ទេ លុះត្រាតែនៅពេលផ្ទេរ bootloader ទៅក្នុងម៉ាយក្រូកុនត្រូល័រថ្មី។ Burn Bootloader ទៅក្នុងម៉ាយក្រូកុនត្រូល័រ។

Help Menu

គេអាចប្រើប្រាស់ menu នេះដើម្បីទៅកាន់សៀវភៅ និងឯកសារយោងផ្សេងៗពីការប្រើប្រាស់ Arduino IDE។



រូបភាព ៣.១៤ ៖ Help menu

៣.១.៤ Arduino Shields

Arduino shield ជា board ដែលអាចតភ្ជាប់បន្ថែមទៅនឹង Arduino board ដើម្បីបង្កើនមុខងាររបស់វា ដូចជា បញ្ជាម៉ូទ័រ តភ្ជាប់ទៅនឹង network ជាដើម សូមមើលរូបភាព ៣.១៥។ ឧទាហរណ៍ Ethernet Shield អាចឲ្យ Arduino តភ្ជាប់ទៅនឹង ethernet network បាន ដែលអាចឲ្យវាផ្ញើ ឬ ទទួលព័ត៌មានពីអ៊ីនធឺណេត អាចបញ្ជាពីចម្ងាយដោយប្រើប្រាស់វែបសាយ ឬ App ជាដើម។

រូបភាព ៣.១៥៖ Ethernet shield (ខាងលើ) តភ្ជាប់ទៅនឹង Arduino UNO

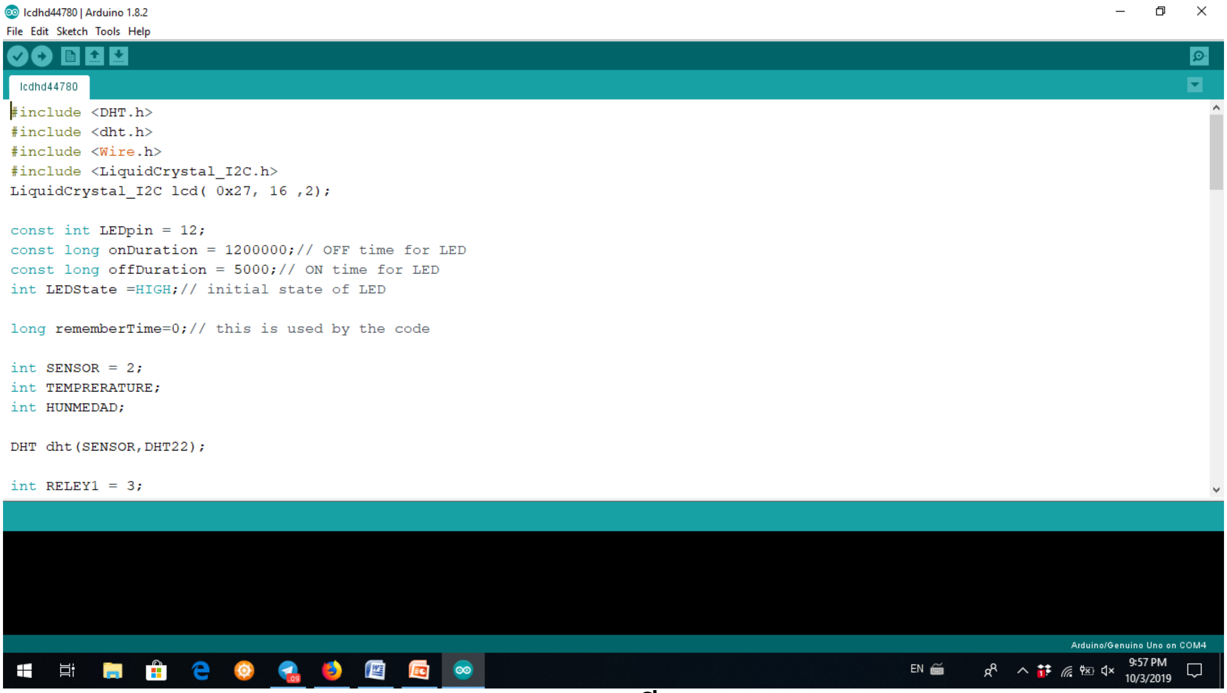
៣.១.៥ Serial Communication

Serial Communication ប្រើប្រាស់ត្រូវបានគេប្រើប្រាស់ដើម្បីទំនាក់ទំនងរវាង Arduino និងកុំព្យូទ័រ ក៏ដូចជា Arduino ផ្សេងៗទៀត វាអាចជា UART, I2C ឬ SPI។ វាមានគុណសម្បត្តិជាង parallel communication ដោយសារប្រើប្រាស់ខ្សែតភ្ជាប់ចំនួនតិចជាង ឧទាហរណ៍ UART ដែលប្រើប្រាស់ USB មានខ្សែតែ២ខ្សែទេ មួយសម្រាប់ផ្ញើព័ត៌មាន និងមួយទៀតសម្រាប់ទទួលព័ត៌មាន។

៣.២ ការសរសេរកម្មវិធីសម្រាប់បញ្ជា Arduino

ការសរសេរកម្មវិធីសម្រាប់ Arduino គេប្រើប្រាស់ Arduino IDE ដូចបង្ហាញនៅក្នុងរូបភាព ៣.១៦។ គេប្រើភាសាសរសេរកម្មវិធីកុំព្យូទ័រប្រភេទ C/C++ ដូចបង្ហាញក្នុងរូបភាព។ អ្នកដែលមានចំណេះជាមុនពីភាសា C/C++ មានភាពងាយស្រួលយល់ sketch របស់ Arduino។

៣.២.១ ការសរសេរកម្មវិធីប្រើប្រាស់ Arduino IDE



រូបភាព ៣.១៦ ៖ ការប្រើប្រាស់ Arduino IDE ដើម្បីសរសេរ sketch របស់ Arduino

ជាធម្មតាគ្រោងឆ្អឹងកម្មវិធី (sketch) របស់ Arduino មាន function ពីរគឺ setup និង loop មើលរូបភាព ៣.១៧។ កូដដែលនៅក្នុង setup ប្រាប់ Arduino ឲ្យធ្វើការងារមួយចំនួននៅពេលបើកដំបូង ចំណែកឯនៅក្នុង loop វិញ វានឹងដំណើរការចុះឡើងៗ។

```
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

រូបភាព ៣.១៧ ៖ គ្រោងឆ្អឹងកូដកម្មវិធីរបស់ Arduino

រូបភាព ៣.១៨ ជាកូដកម្មវិធីបញ្ជា Arduino ឲ្យបញ្ចេញភ្លើង LED នៅលើ board របស់ Arduino។ ជួរដេកដែលចាប់ផ្តើមដោយ "//" ជា comment ដែល Arduino មិនរាប់បញ្ចូលនៅពេលបំលែងទៅជាកូដម៉ាស៊ីនទេ នៅពេលចុចលើ icon Verify/Compile ឬ Upload នៅលើ tool bar។ វាគ្រាន់តែជាការកត់ចំណាំរបស់អ្នកសរសេរកម្មវិធីកុំឲ្យភ្លេចនៅពេលយកវាមកមើលវិញ ក្រោយពេលទុកកូដចោលយូរ។ ការប្រើប្រាស់ comment ដើម្បីចំណាំបន្ទាត់ ឬ ផ្នែករបស់កូដ ជាទម្លាប់ល្អសម្រាប់អ្នកសរសេរកម្មវិធី ដោយសារក្រោយពេលទុកកូដចោលយូរ យើងអាចភ្លេចថាបន្ទាត់នេះ ឬ ផ្នែកនេះសម្រាប់ធ្វើអ្វីជាដើម។

នៅក្នុង function setup មានកូដដូចខាងក្រោមនេះ ៖

```
pinMode( LED_BUILDIN, OUTPUT );
```

ដែលប្រាប់ Arduino ថាចង់ប្រើប្រាស់ LED នៅលើboard ដែល pin របស់វាតំណាងដោយតម្លៃថេរ LED_BUILDIN សម្រាប់បញ្ចេញព័ត៌មាន (OUTPUT)។

```
// the setup function runs once when you press reset or power the board
void setup( ) {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode( LED_BUILTIN, OUTPUT );
}
// the loop function runs over and over again forever
void loop( ) {
  digitalWrite( LED_BUILTIN, HIGH );   // turn the LED on ( HIGH is the voltage level )
  delay( 1000 );                       // wait for a second
  digitalWrite( LED_BUILTIN, LOW );    // turn the LED off by making the voltage LOW
  delay( 1000 );                       // wait for a second
}
```

រូបភាព ៣.១៨ ៖ កូដបញ្ជា Arduino ឲ្យបញ្ជាភ្លើងរបស់ LED

កូដដែលនៅក្នុង loop( ) វានឹងរត់ចុះឡើង ដដែលៗ។

```
digitalWrite( LED_BUILTIN, HIGH );
```

បញ្ចេញចរន្តទៅ LED នៅលើ board របស់ Arduino (LED_BUILDIN) ធ្វើឲ្យវាភ្លឺ HIGH។

delay(1000);

បញ្ជាឲ្យ Arduino រង់ចាំ 1000 ms (1 វិនាទី) មុននឹងបន្តទៅកូដនៅជួរដេកបន្ទាប់។

digitalWrite(LED_BUILTIN, LOW);

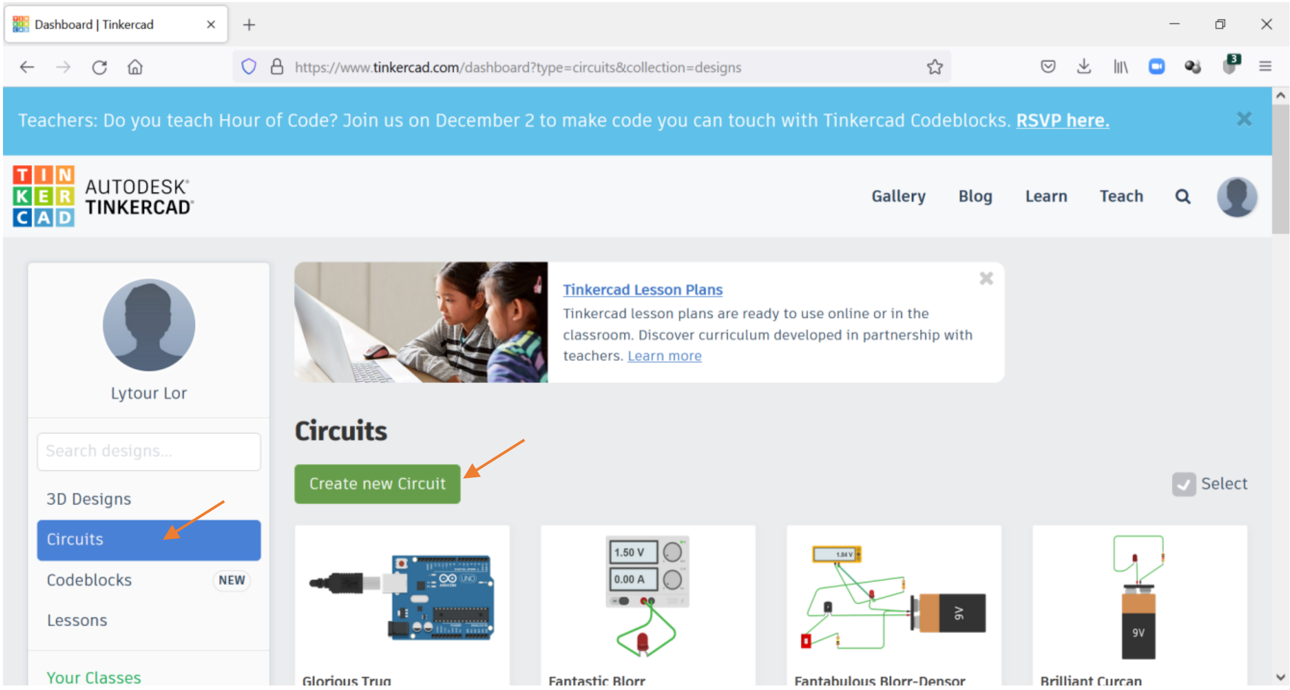
បិទ LED នៅលើ board របស់ Arduino ឲ្យឈប់ភ្លឺ។

សរុបមកកូដនៅក្នុង loop( ) បញ្ជាឲ្យ LED ភ្លឺ រួចរង់ចាំ១វិនាទី បន្ទាប់បិទឲ្យឈប់ភ្លឺ រួចរង់ចាំ១វិនាទី ហើយធ្វើបែបនេះដដែលៗ ធ្វើឲ្យ LED ភ្លឺភ្លឺបភ្លែតៗ។

៣.២.២ ការសរសេរកម្មវិធី Arduino ប្រើប្រាស់ TinkerCAD

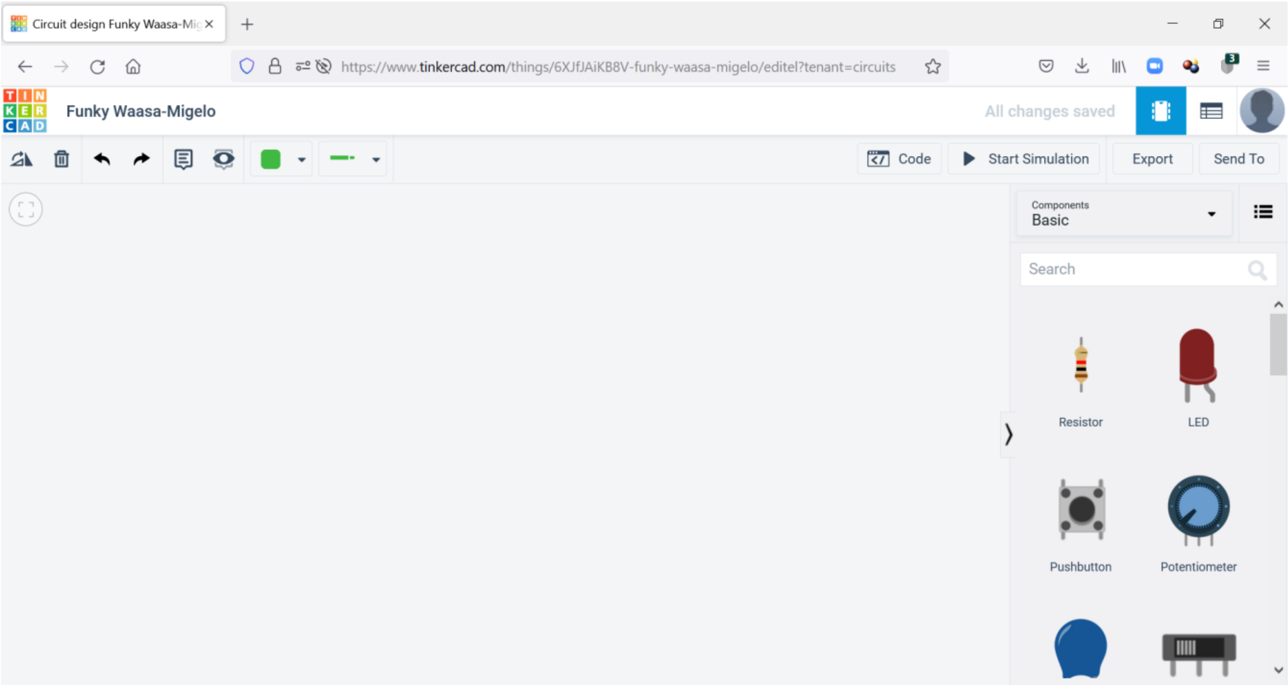
ក្រៅពីប្រើប្រាស់ Arduino IDE និងឧបករណ៍ពិតប្រាកដដើម្បីសរសេរកម្មវិធី Arduino គេអាចប្រើប្រាស់ simulator ដូចជា Tinkercad សម្រាប់សរសេរកម្មវិធីរបស់ Arduino និងដំណើរការដោយពុំត្រូវការ hardware ពិតប្រាកដទេ។ Tinkercad អាចប្រើប្រាស់ក្នុងការរៀនគ្រឿងអេឡិកត្រូនិក និងសរសេរកម្មវិធីសម្រាប់ Arduino បានមួយចំនួនធំ។ ដើម្បីប្រើប្រាស់វាសូមធ្វើដូចខាងក្រោម ៖

១. Login ចូលវេបសាយ https://www.tinkercad.com (គេអាចប្រើប្រាស់ Google account ដើម្បី log in ចូលបានដែរ)។ ក្រោយពី log in ចូលរួចហើយ ទំព័រនឹងបង្ហាញស្រដៀងក្នុងរូបភាព ៣.១៩ ខាងក្រោម។



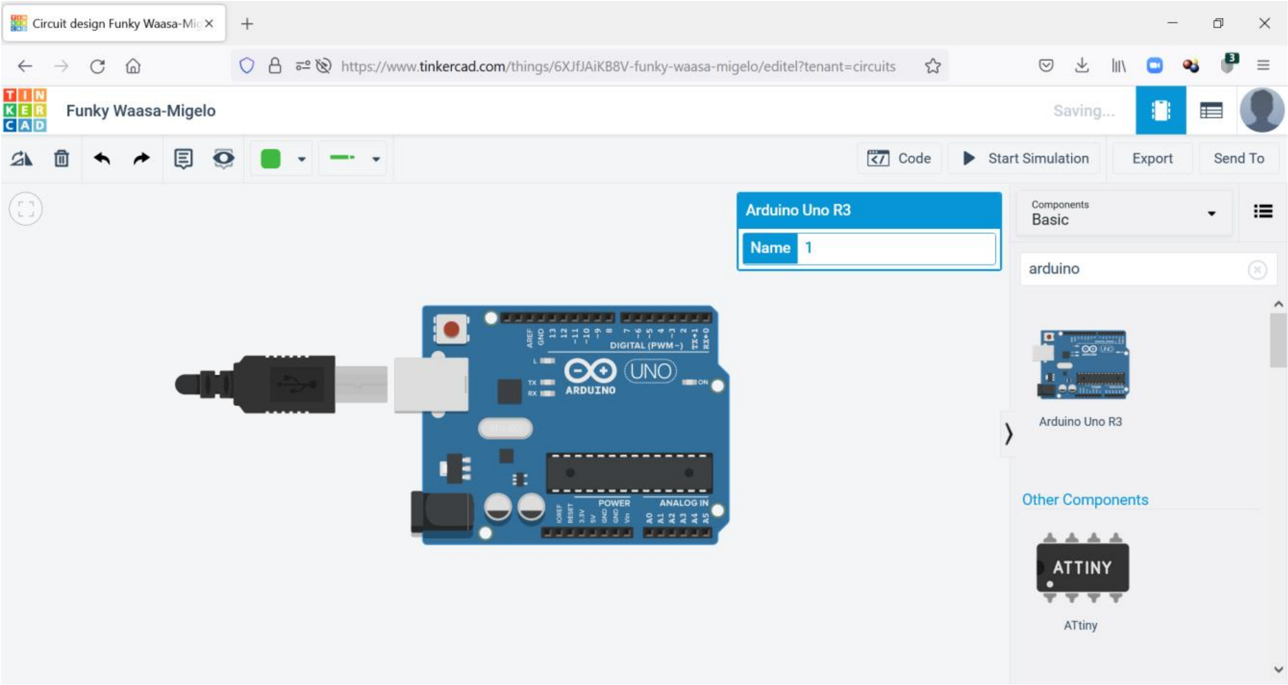
រូបភាព ៣.១៩ ៖ ទំព័រ TinkerCAD ក្រោយពី log in

ចុចលើ Circuits -> Create new Circuit វានឹងបង្ហាញទំព័រដូចក្នុងរូបភាព ៣.២០។

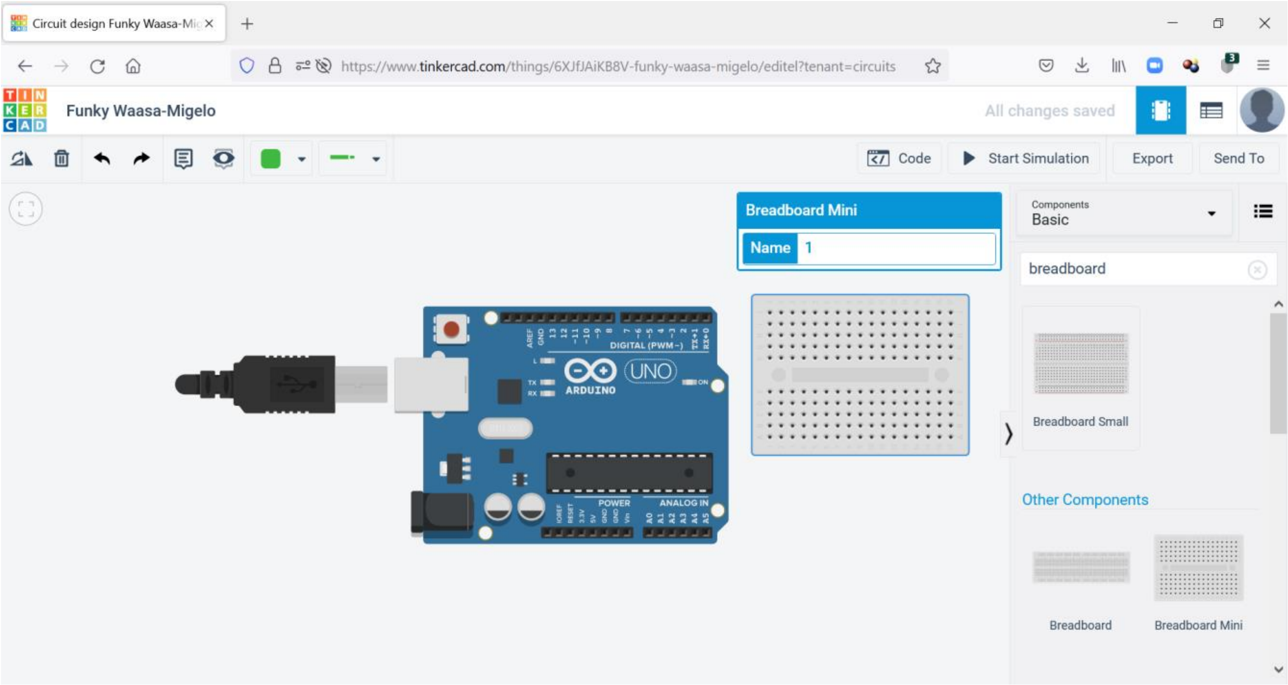


រូបភាព ៣.២០ ៖ ទំព័រ TinkerCAD ក្រោយពីចុចលើ Create new Circuit

នៅកន្លែង search វាយពាក្យ Arduino វានឹងបង្ហាញ board Arduino Uno R3។ ចាប់ទាញនឹងទម្លាក់វាមកលើទំព័រទទេនៅខាងឆ្វេងដៃ គេនឹងឃើញ board Arduino Uno បង្ហាញខ្លួននៅលើទំព័រនេះ ដូចក្នុងរូបភាព ៣.២១។

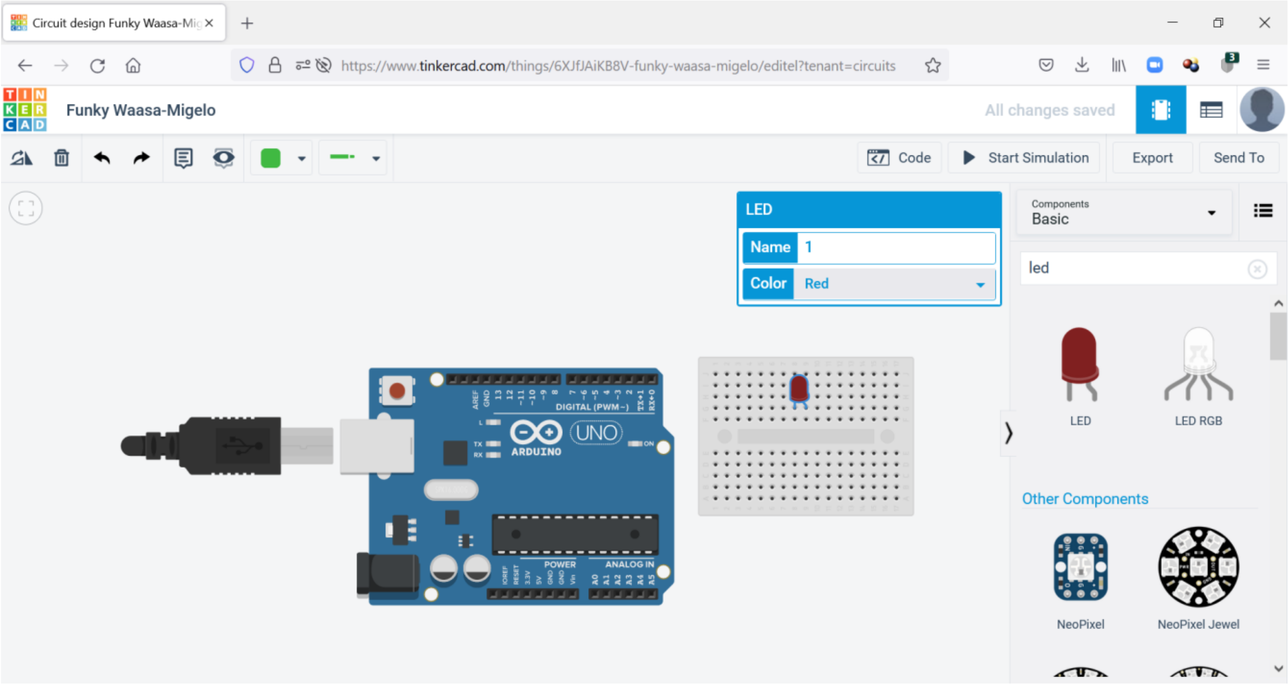


រូបភាព ៣.២១ ៖ ការបន្ថែម Arduino board



រូបភាព ៣.២២ ៖ ការបន្ថែម Breadboard Mini

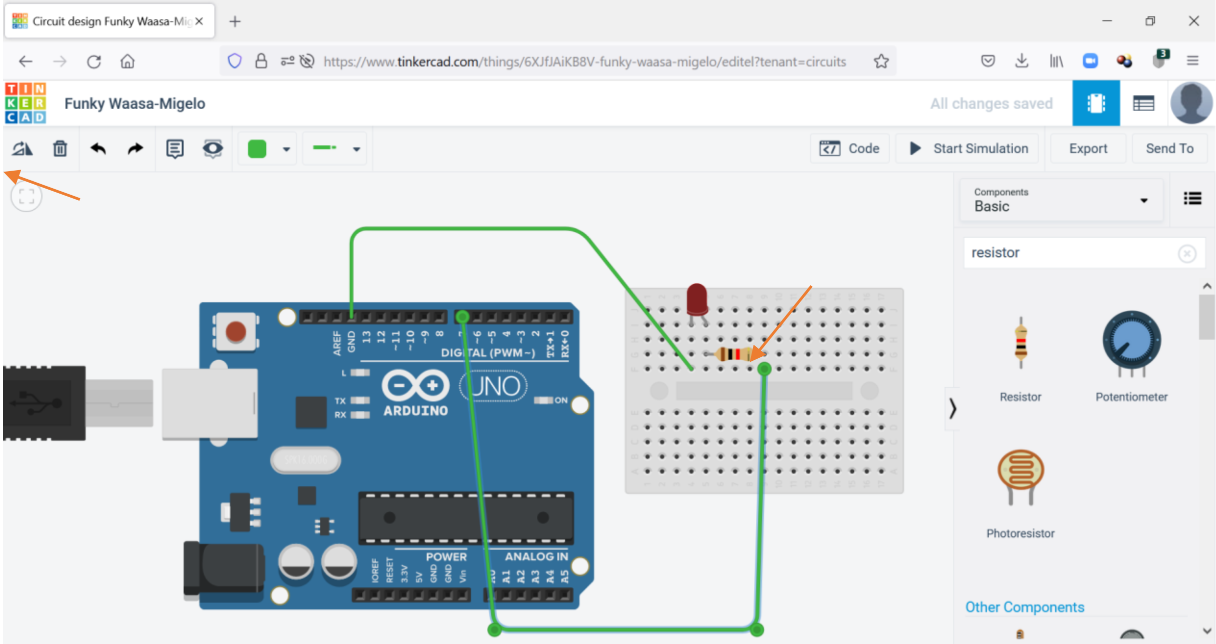
នៅកន្លែង search វាយពាក្យ breadboard និងចុច Enter នៅលើ keyboard រួចហើយចាប់ទាញ ដើម្បីបន្ថែម Breadboard Mini ទៅដាក់ក្បែរ Arduino Uno ដូចបង្ហាញក្នុងរូបភាព ៣.២២។



រូបភាព ៣.២៣ ៖ ការបន្ថែម LED ទៅលើ Breadboard Mini

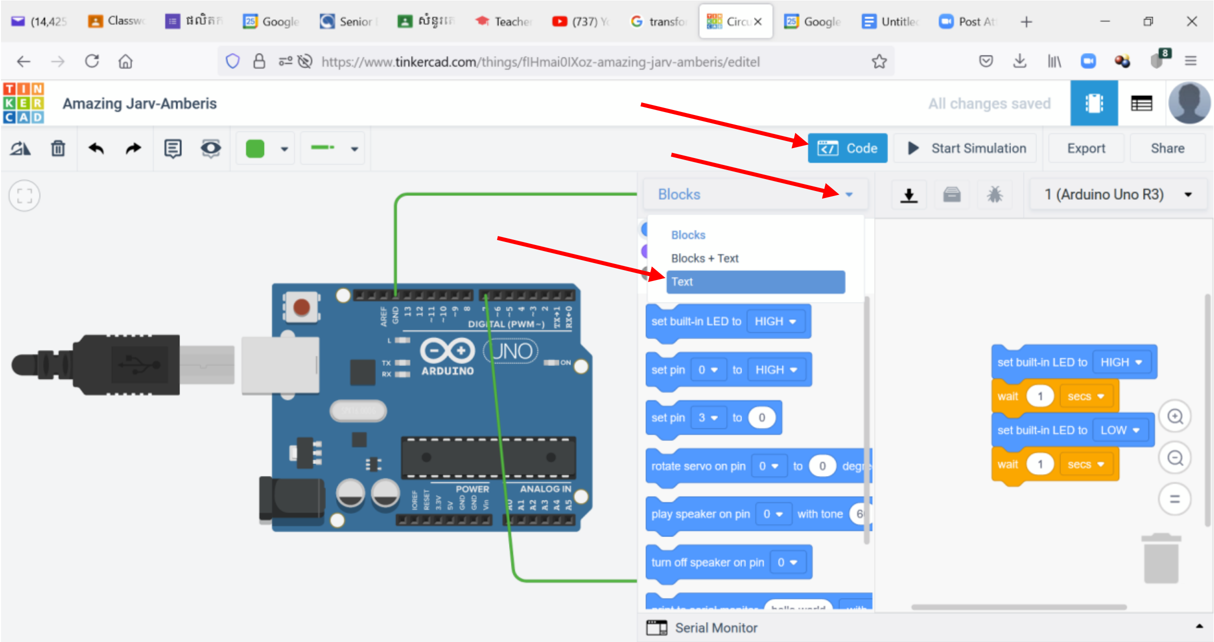
នៅកន្លែង search វាយពាក្យ led រួចចុច Enter នៅលើ keyboard បន្ទាប់មកបន្ថែម led ទៅលើ Breadboard Mini ដូចបង្ហាញក្នុងរូបភាព ៣.២៣។

ធ្វើដូចគ្នា ដើម្បីបន្ថែមរ៉េស៊ីស្តង់ (resistor) ទៅលើ Breadboard Mini។ បង្វិលរ៉េស៊ីស្តង់ឲ្យដេក ដោយចុចលើ icon នៅផ្នែកខាងលើរបស់ទំព័រ។ ចុចលើ mouse pointer រួចទាញ ដើម្បីតភ្ជាប់កាតូតរបស់ LED ទៅនឹង GND របស់ Arduino និងចុងខាងស្តាំដៃរបស់រ៉េស៊ីស្តង់ទៅនឹង digital pin 7 របស់ LED សូមមើលរូបភាព ៣.២៤។



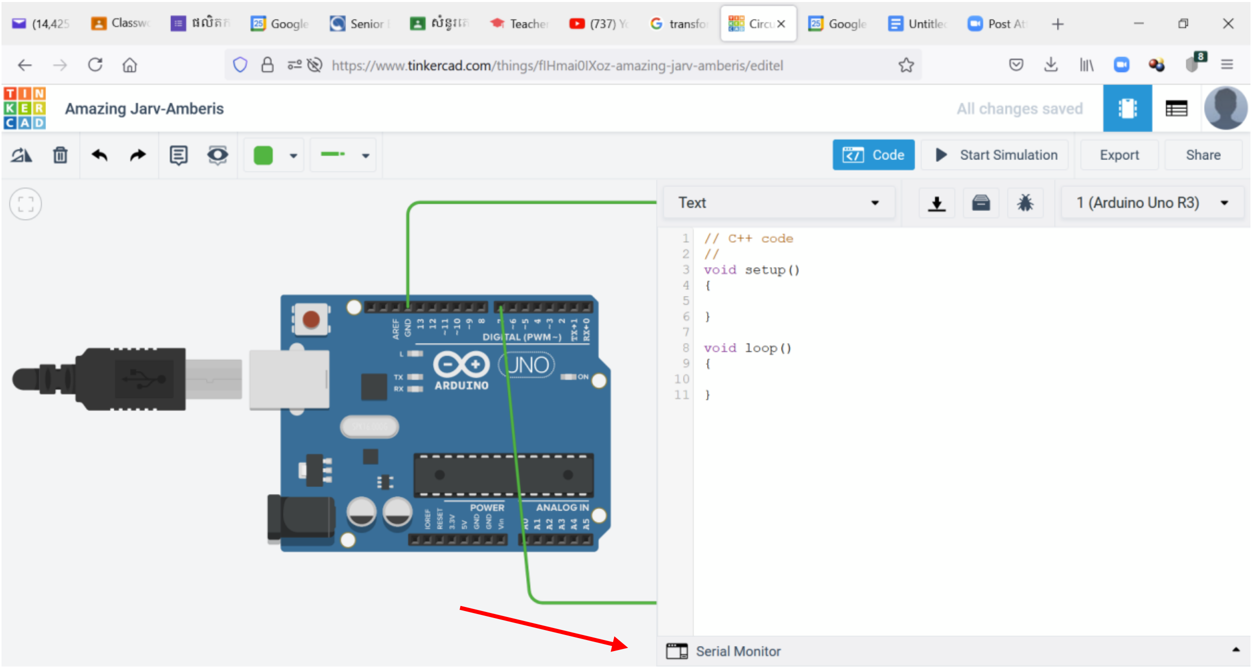
រូបភាព ៣.២៤ ៖ សៀគ្វីក្រោយពេលតភ្ជាប់រួចរាល់

២. ចុចលើ Code បន្ទាប់ចុចលើ Blocks រួចជ្រើសរើសយក Text។



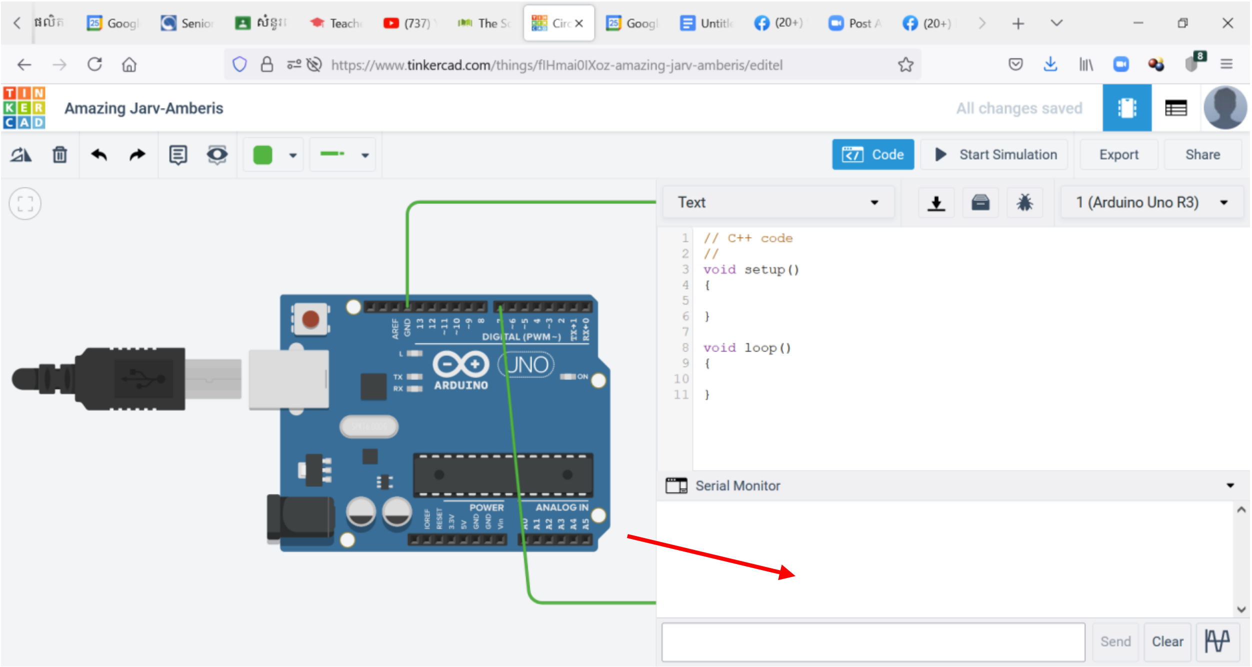
រូបភាព ៣.២៥ ៖ ការបង្ហាញ code editor

៣. បន្ទាប់មកចុចលើ Continue ដើម្បីបង្ហាញកន្លែងសរសេរ code ដូចក្នុងរូបភាព ៣.២៦ ខាងក្រោម។



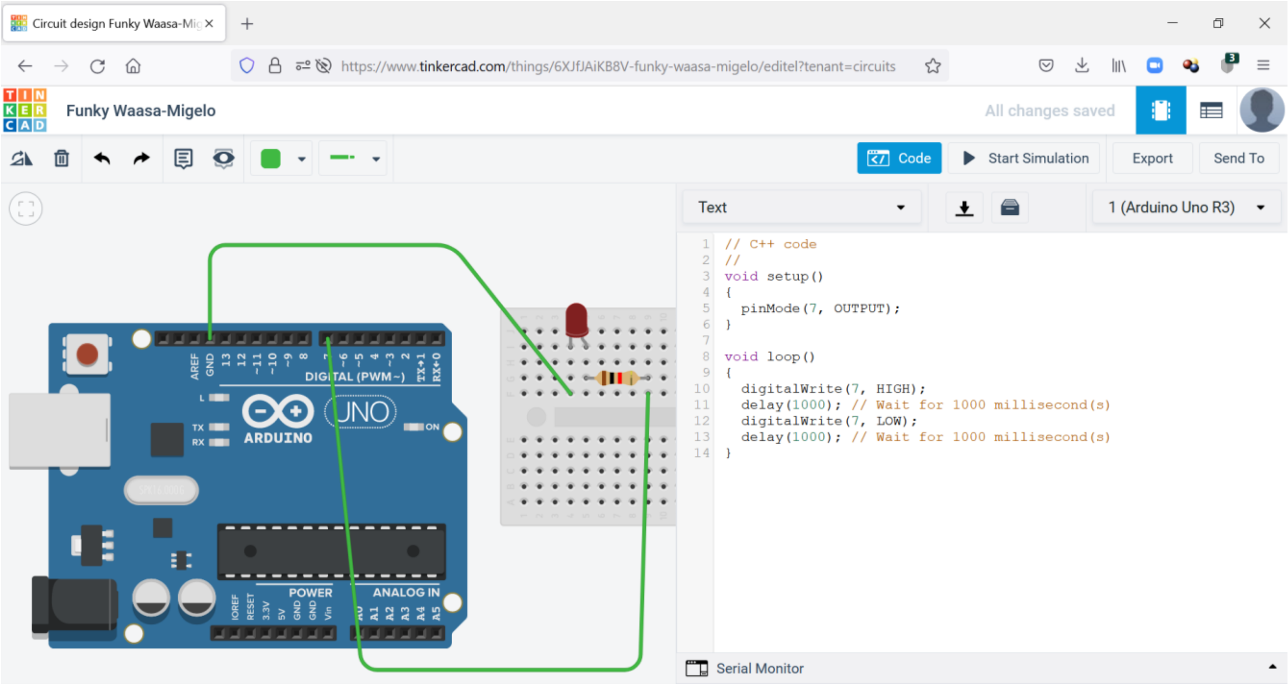
រូបភាព ៣.២៦ ៖ ទំព័របង្ហាញ code editor

៤. ចុចលើ Serial Monitor នៅខាងក្រោមដើម្បីបង្ហាញវា ដូចក្នុងរូបភាព ៣.២៧ ខាងក្រោម។



រូបភាព ៣.២៧ ៖ ការបង្ហាញ Serial Monitor

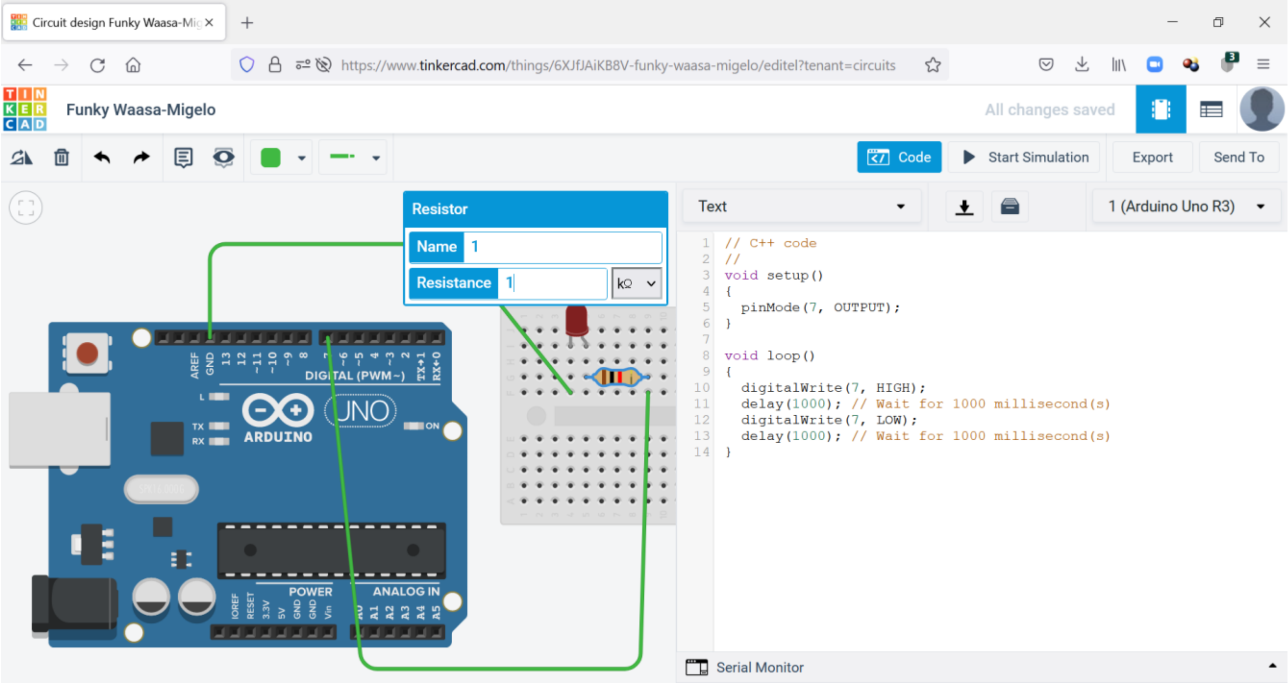
៥. Copy និង Paste កូដសម្រាប់ធ្វើឲ្យ LED ភ្លឺភ្លឹបភ្លែត ទៅក្នុងកន្លែងសរសេរកូដ។ ដូចបង្ហាញក្នុងរូបភាព ៣.២៨ ខាងក្រោម។



រូបភាព ៣.២៨ ៖ TinkerCAD បង្ហាញកូដធ្វើឲ្យ LED ភ្លឺភ្លឹបភ្លែត

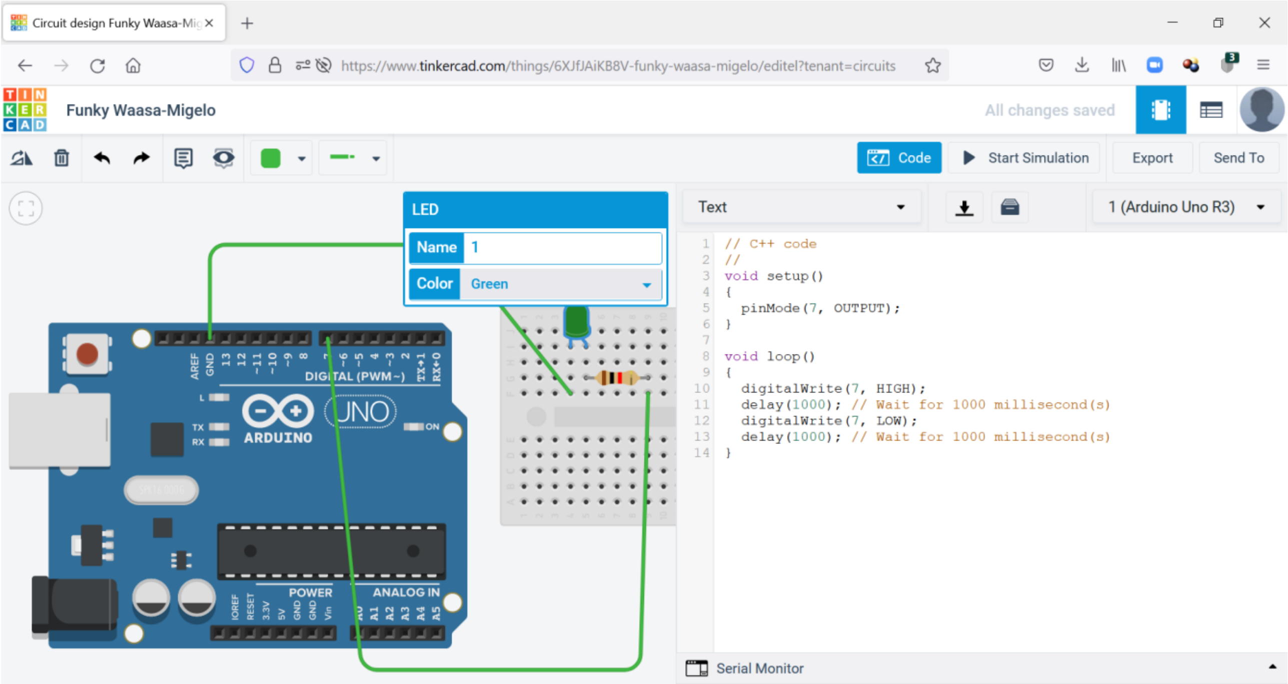
៦. ចុចលើ Start Simulation ដើម្បីដំណើរការវា គេនឹងឃើញ LED ភ្លឺភ្លឹបភ្លែតៗ រៀងរាល់មួយវិនាទី នៅក្នុង TinkerCAD។

គេអាចប្តូរតម្លៃរ៉េស៊ីស្តង់ ដោយចុចលើវា រួចហើយប្តូរតម្លៃ ដោយវាយតម្លៃថ្មី និងជ្រើសរើសខ្នាត ដូចក្នុងរូបភាព ៣.២៩។



រូបភាព ៣.២៩ ៖ ការប្តូរតម្លៃរ៉េស៊ីស្តង់

ដើម្បីប្តូរពណ៌ LED ចុចលើវា ហើយជ្រើសរើសពណ៌ដូចបង្ហាញក្នុងរូបភាព ៣.៣០។



រូបភាព ៣.៣០៖ ការប្តូរពណ៌ LED

៣.២.៣ ប្រភេទទិន្នន័យ (Data Type) ភាសាកុំព្យូទ័រ C របស់ Arduino

ភាសាកុំព្យូទ័រ C របស់ Arduino ស្គាល់ប្រភេទទិន្នន័យភាគច្រើនរបស់ ANSI C មានតែករណីលើកលែងមួយចំនួនប៉ុណ្ណោះ។ នៅពេលគេតាងអញ្ញត្តិ គេត្រូវប្រាប់ថាវាជាប្រភេទទិន្នន័យអ្វី? ឧទាហរណ៍ជាលេខគត់ លេខមានក្បៀស ឬ អក្សរ។ល។ ប្រភេទទិន្នន័យផ្សេងគ្នា ប្រើប្រាស់ memory ទំហំខុសគ្នា ហេតុនេះហើយការជ្រើសរើសប្រភេទទិន្នន័យដែលត្រូវនឹងការប្រើប្រាស់មានសារៈសំខាន់ណាស់ ដោយសារវាធ្វើឲ្យការប្រើប្រាស់ memory មានប្រសិទ្ធភាព ធ្វើឲ្យកម្មវិធីដំណើរការលឿន និងមិនមានបញ្ហា។ ប្រភេទទិន្នន័យសំខាន់មានបង្ហាញខាងក្រោម។

ប្រភេទទិន្នន័យ	ប្រវែងគិតជា Byte	ចន្លោះតម្លៃ
boolean	1	មានតែតម្លៃពីរគឺ true (ពិត) និង false (មិនពិត)
char	1	–128 ទៅ +127
unsigned char	1	0 ទៅ 255
byte	1	0 ទៅ 255
int	2	–32,768 ទៅ 32,767
unsigned int	2	0 ទៅ 65,535
word	2	0 ទៅ 65,535
long	4	–2,147,483,648 ទៅ 2,147,483,647
unsigned long	4	0 ទៅ 4,294,967,295
float	4	–3.4028235E+38 ទៅ 3.4028235E+38
double	4	–3.4028235E+38 ទៅ 3.4028235E+38
void	0	ប្រើប្រាស់នៅពេល function មិនបញ្ចេញតម្លៃ

៣.២.៤ ការសម្រេចចិត្តក្នុងភាសាកុំព្យូទ័រ C

សមត្ថភាពពិតប្រាកដរបស់ microcontroller គឺវាអាចអានទិន្នន័យ (អាចពីសេនស័រ) ហើយធ្វើសកម្មភាពអាស្រ័យលើតម្លៃរបស់ទិន្នន័យនោះ។ វាអាចធ្វើការសម្រេចចិត្តដោយពិនិត្យមើលលើព័ត៌មានដែលផ្តល់ទៅឲ្យវា។ គេអាចប្រើប្រាស់ការប្រៀបធៀបតម្លៃ ដូចជា ធំជាង តូចជាង ស្មើគ្នា ឬ មិនស្មើគ្នាជាដើម រួចវាយតម្លៃថាវាពិត ឬ មិនពិត។ ខាងក្រោមនេះជាសញ្ញាប្រៀបធៀបដែលនិយមប្រើប្រាស់។

> ធំជាង

>=	ធំជាង ឬ ស្មើ
<	តូចជាង
<=	តូចជាង ឬ ស្មើ
==	ស្មើគ្នា
!=	មិនស្មើគ្នា

ឧទាហរណ៍

5 > 4 // Logic true	ពិត
5 < 4 // logic false	មិនពិត
5 == 4 // logic false	មិនពិត
5 != 4 // logic true	ពិត

គេអាចប្រើប្រាស់វាជាមួយអញ្ញត្តិបានផងដែរ ឧទាហរណ៍ ប្រសិនបើ a = 5 និង b = 4 ដូច្នេះ

a > b // Logic true	ពិត
a < b // logic false	មិនពិត
a == b // logic false	មិនពិត
a != b // logic true	ពិត

ការប្រើប្រាស់ if

ខាងក្រោមនេះជារូបមន្តសម្រាប់ប្រើប្រាស់ if

```
if ( expression1 ពិត ) {
        // ដំណើរការកូដក្នុង { }
}
```

សូមពិនិត្យមើលផ្នែកនៃកូដខាងក្រោមនេះ។

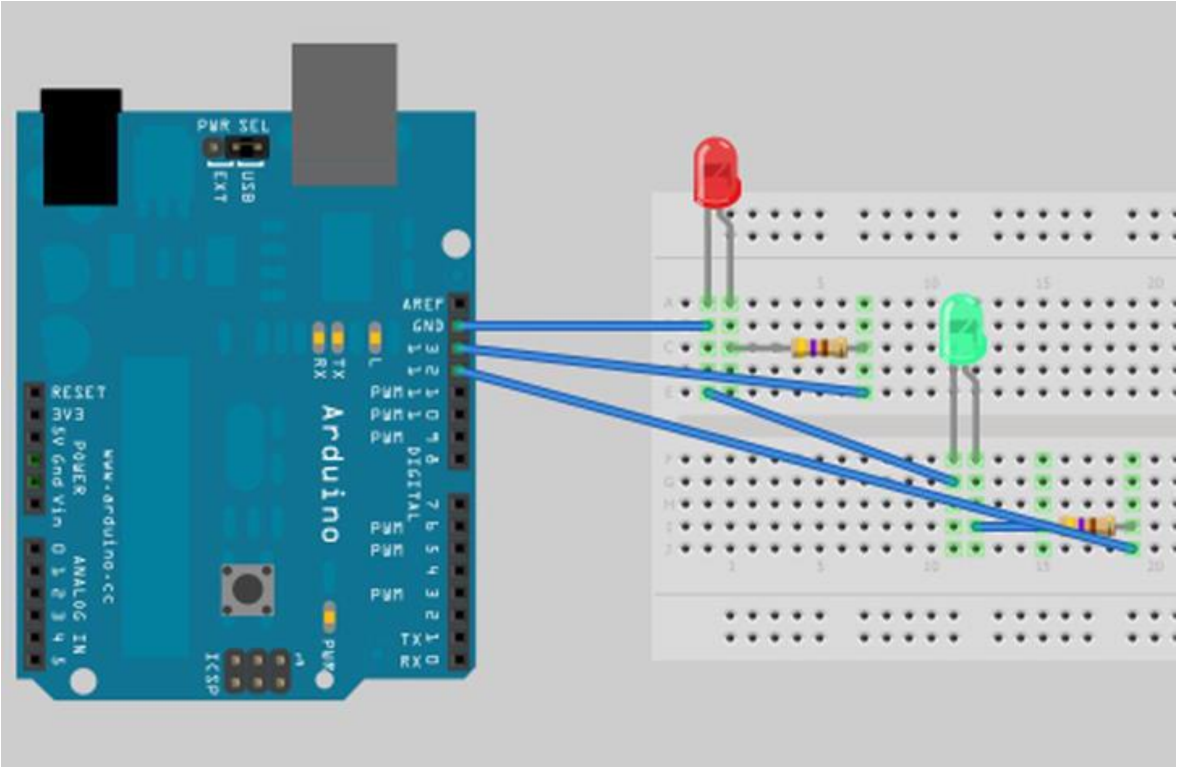
```
int b = 10;

if ( b < 20 ) {
```

```
    b = doSomethingNeat( );
}
  doSomethingElse( b );
```

កូដនេះចាប់ផ្តើមដោយតាង b ជាលេខគត់មានតម្លៃ 10។ បន្ទាប់មក វាធ្វើការសម្រេចចិត្តដោយពិនិត្យមើលតម្លៃ b ដោយប្រើប្រាស់ if។ តើ b តូចជាង 20 ដែរឬទេ? ក្នុងករណីនេះ b តូចជាង 20 ហេតុនេះ កូដ b = doSomethingNeat( ); ដំណើរការ ដោយផ្លាស់ប្តូរតម្លៃ b ស្មើនឹងតម្លៃផ្តល់ឲ្យដោយ function doSomethingNeat( )។ បន្ទាប់មកទៀតកូដ doSomethingElse(b); ដំណើរការ។

កម្មវិធីធ្វើឲ្យអំពូល LED ភ្លឺភ្លឹបភ្លែតកែសម្រួល

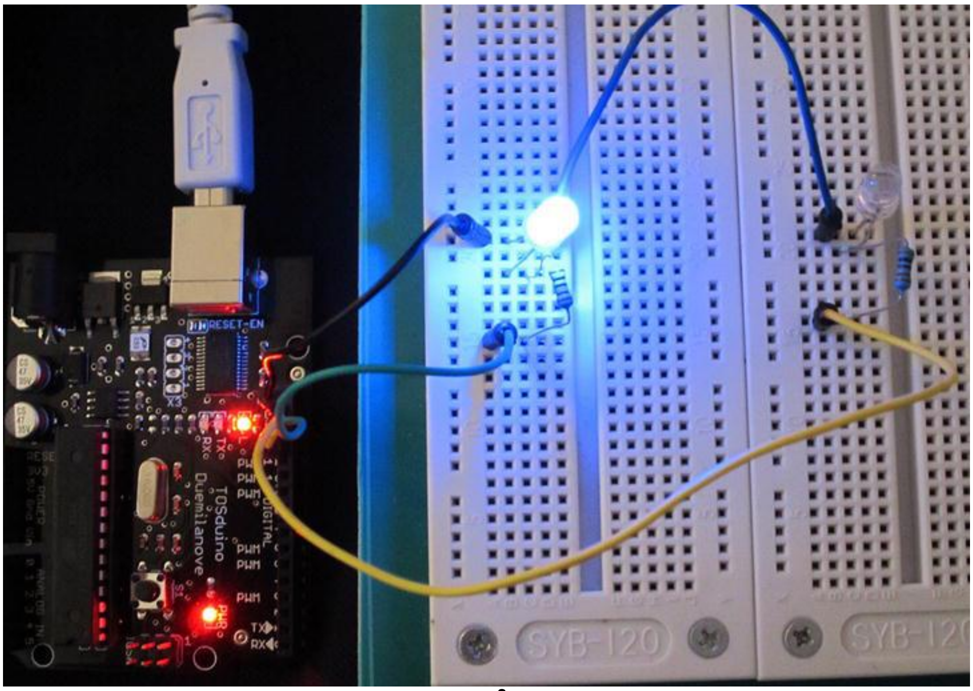
កម្មវិធីនេះនឹងប្រើប្រាស់អំពូល LED ពីរ រ៉េស៊ីស្តង់ពីរ និងខ្សែចម្លងមួយចំនួនទៀត ដូចបង្ហាញក្នុង រូបភាព ៣.៣១។ នៅពេលដំណើរការ វានឹងធ្វើឲ្យអំពូល LED ទី១ ភ្លឺ ១ វិនាទី បន្ទាប់មករលត់ ហើយភ្លឺអំពូលមួយទៀត។



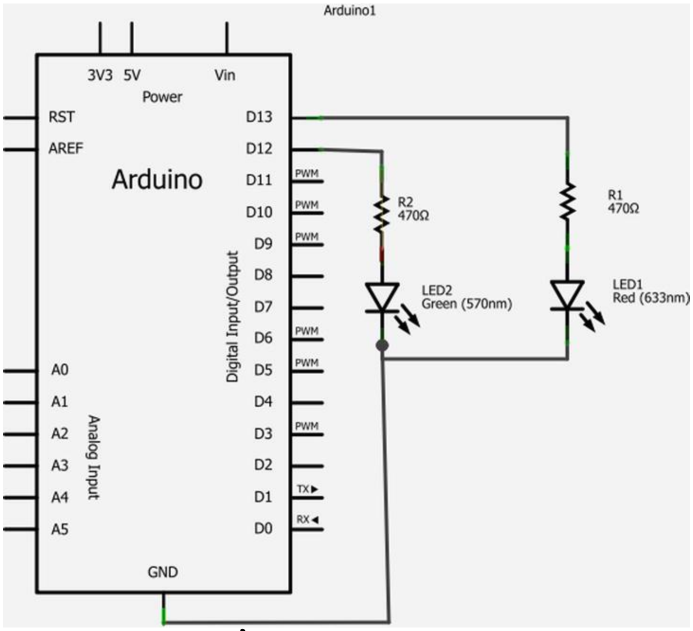
រូបភាព ៣.៣១ ៖ ការប្រើប្រាស់ Arduino ដើម្បីបញ្ជាអំពូល LED ពីរ

សៀគ្វីនេះតភ្ជាប់ខ្សែចម្លងមួយពី digital pin 13 របស់ Arduino ទៅកាន់រ៉េស៊ីស្តង់។ ចុងម្ខាងទៀតនៃរ៉េស៊ីស្តង់តភ្ជាប់ទៅនឹងអាណូតរបស់ LED (ជើងវែង) ចំណែកឯកាតូតតភ្ជាប់ទៅនឹង GND របស់ Arduino។

របៀបតភ្ជាប់ LED ទី២ ស្រដៀងនឹងទី១ដែរ គ្រាន់តែប្រើប្រាស់ digital pin 12 ជំនួសវិញ។ តើគេគួរជ្រើសរើសតម្លៃរបស់រ៉េស៊ីស្តង់ប៉ុន្មាន? វាអាស្រ័យទៅនឹង LED ដែលគេប្រើប្រាស់ ប៉ុន្តែបន្ទុកអតិបរមារបស់ Arduino pin មិនគួរលើសពី 20 mA ទេ។ ចំណែកឯ LED វិញប្រើប្រាស់ចរន្តអតិបរមា 5 mA ហេតុនេះវាមិនលើសសមត្ថភាពរបស់ pin ទេ។ ដើម្បីគណនារ៉េស៊ីស្តង់ គេប្រើប្រាស់រូបមន្តច្បាប់អូម រ៉េស៊ីស្តង់ = តង់ស្យុង / អាំងតង់ស៊ីតេចរន្ត។ ដោយសារតង់ស្យុងផ្តល់ដោយ pin មានតម្លៃ 5 V និងអាំងតង់ស៊ីតេអតិបរមា 20 mA ឬ 0.020 A ហេតុនេះតម្លៃរបស់រ៉េស៊ីស្តង់គឺ 250 អូម។ នៅពេលគេប្រើប្រាស់តម្លៃរ៉េស៊ីស្តង់នេះ pin បញ្ចេញចរន្តអតិបរមា ដែលអាចបង្កបញ្ហាឆេះខូច LED ហេតុនេះដើម្បីមានសុវត្ថិភាពតម្លៃរបស់រ៉េស៊ីស្តង់ត្រូវបានបង្កើនរហូតដល់ 470 អូម។ គេអាចចាប់ផ្តើមសាកល្បងពីតម្លៃរ៉េស៊ីស្តង់ធំ ហើយពិនិត្យមើលពន្លឺរបស់ LED ប្រសិនបើខ្សោយពេក គេអាចសាកល្បងប្រើប្រាស់រ៉េស៊ីស្តង់តូចជាងនេះបន្តិច រហូតដល់ LED ភ្លឺល្អ។ ពេលបន្ថយតម្លៃរ៉េស៊ីស្តង់ LED ភ្លឺជាងមុន ដោយសារអាំងតង់ស៊ីតេចរន្តដែលរត់កាត់វាធំជាងមុន។ សម្រាប់ LED ភាគច្រើន តម្លៃរ៉េស៊ីស្តង់ពី 470 អូម ទៅ 1,000 អូម អាចប្រើប្រាស់បាន។ រូបភាព ៣.៣២ បង្ហាញនៅពេល Arduino ដំណើរការកូដក្នុងរូបភាព ៣.៣៤ និងការតភ្ជាប់សៀគ្វីក្នុងរូបភាព ៣.៣៣។



រូបភាព ៣២ ៖ LED ផ្លាស់ប្តូរវេនគ្នាភ្លឺនៅពេលដំណើរការ



រូបភាព ៣.៣៣ ៖ ការតភ្ជាប់របស់សៀគ្វី LED ផ្លាស់ប្តូរវេនគ្នាភ្លឺ

ខាងក្រោមនេះជាកូដរបស់កម្មវិធី

```
/*
Alternate Blink
Turns on one LED on for 1 second while the other is off, then reverses the LEDs for 1
second,
repeatedly.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led1 = 13;
int led2 = 12; // This is for the second LED
// the setup routine runs once when you press reset:
void setup( ) {
  // initialize the digital pin as an output.
  pinMode( led1, OUTPUT );
  pinMode( led2, OUTPUT );
}
```

```
// the loop routine runs over and over again forever:
void loop( ) {
  digitalWrite( led1, HIGH ); // turn the LED on ( HIGH is the voltage level )
  digitalWrite( led2, LOW ); // turn the LED off by making the voltage LOW
  delay( 1000 ); // wait for a second
  digitalWrite( led1, LOW ); // turn the LED off by making the voltage LOW
  digitalWrite( led2, HIGH ); // turn the LED on ( HIGH is the voltage level )
  delay( 1000 ); // wait for a second
}
```

រូបភាព ៣.៣៤ ៖ កូដសម្រាប់ឲ្យអំពូល LED ពីរប្តូរវេនគ្នាភ្លឺ

កម្មវិធីចាប់ផ្តើមដោយមាន comment ជាច្រើនជួរប្រើប្រាស់ទម្រង់ /* ……… */ និង //……។ បន្ទាប់មកទៀតមានតាងអញ្ញត្តិពីរមានប្រភេទទិន្នន័យជាលេខគត់ (int)។ អញ្ញត្តិទាំងពីរតាងតម្លៃ digital pin ដែល LED ទាំងពីរតភ្ជាប់ជាមួយ Arduino។ បន្តមកទៀត កម្មវិធីដំណើរការ function setup( ) ដើម្បីចាប់ផ្តើមដំណើរការកម្មវិធី ដោយជ្រើសរើស pin ទាំងពីរជា output ដោយប្រើប្រាស់ function pinMode( )។ មានន័យថា នៅពេល function digitalWrite( ) ដំណើរការ វានឹងបញ្ចេញចរន្តអគ្គីសនីមានតង់ស្យុង +5 V នៅពេល pin mode មានតម្លៃ HIGH។ នៅពេល pin mode ប្តូរទៅជា LOW វិញ វានឹងឈប់ផ្តល់ចរន្តអគ្គីសនីទៅ។ បន្តទៅទៀតកម្មវិធី នឹងដំណើរ function loop( ) ដែលដំណើរការសារចុះឡើង។ វាចាប់ផ្តើមដោយ ដំណើរការ function digitalWrite( ) សម្រាប់ led1 ដោយប្តូរតម្លៃ pin របស់វាទៅជា HIGH (+5 V)។ បន្ទាប់មកទៀតដំណើរការ function digitalWrite( ) សម្រាប់ led2 ដោយប្តូរតម្លៃរបស់វា LOW (0 V)។ បន្ទាប់មកទៀត វាពន្យាពេល 1 វិនាទី (1,000 មីលីវិនាទី) ដោយដំណើរការ function delay(1000)។ បន្តទៀត កម្មវិធីដំណើរការ function digitalWrite( ) ដើម្បីប្តូរតម្លៃរបស់ led1 និង led2 ឲ្យបញ្ច្រាសពីមុន ហើយពន្យាពេលមិនធ្វើអ្វីរយៈពេល 1 វិនាទីទៀត។ កម្មវិធីនេះធ្វើឲ្យ led1 និង led2 ផ្លាស់ប្តូរវេនគ្នាភ្លឺ និង រលត់ រៀងរាល់ 1 វិនាទីម្តង រហូតដល់គេឈប់ផ្តល់អគ្គីសនីទៅឲ្យ Arduino។

ការកែសម្រួលកម្មវិធីដោយប្រើប្រាស់ if

ឥឡូវនេះ យើងកែសម្រួលកម្មវិធីខាងលើ ដោយប្រើប្រាស់ if វិញ ដោយយើងកែសម្រួលតែ function loop( ) ប៉ុណ្ណោះដូចបង្ហាញខាងក្រោម។

```
// the loop routine runs over and over again forever:
void loop( ) {
```

```
if ( counter % 2 == 1 ) {
  digitalWrite( led1, LOW ); // turn the LED off by making the voltage LOW
  digitalWrite( led2, HIGH ); // turn the LED on ( HIGH is the voltage level )
  delay( 1000 ); // wait for a second
}
if ( counter % 2 == 0 ) {
  digitalWrite( led1, HIGH ); // turn the LED on ( HIGH is the voltage level )
  digitalWrite( led2, LOW ); // turn the LED off by making the voltage LOW
                    delay( 1000 ); // wait for a second
}
counter = counter + 1;
}
```

គេតាងអញ្ញត្តិ counter ដែលមានប្រភេទទិន្នន័យជា long ហើយផ្តល់តម្លៃឲ្យវាស្មើ 0 នៅក្នុងកូដជួរបន្ទាប់ពីតាងអញ្ញត្តិ led1 និង led2។ ការគណនា counter % 2 គឺជាការរកសំណល់នៃផលចែក ដោយយកតម្លៃបច្ចុប្បន្នរបស់ counter ចែកនឹង 2 ហើយរកសំណល់នៃផលចែក ហេតុនេះវាដូចគ្នាទៅនឹងសួរថាតើតម្លៃ counter មានតម្លៃសេស ឬ គូ? សូមពិនិត្យមើលខាងក្រោមនេះ

1 % 2 = 1

2 % 2 = 0

3 % 2 = 1

4 % 2 = 0

5 % 2 = 1

ដោយសារតម្លៃ counter ត្រូវបានបង្កើនម្តង 1 ដោយប្រើប្រាស់កូដ counter = counter + 1 នៅពេលវាដំណើរការ function loop( ) ម្តង វាតេស្តតម្លៃ counter ដែលធ្វើឲ្យ LED ទាំងពីរឆ្លាស់គ្នាប្តូរវេនភ្លឺស្រដៀងទៅនឹងកម្មវិធីមុននេះដែរ។

ការប្រើប្រាស់ if-else

C អាចប្រើប្រាស់ទម្រង់ if មួយទៀត ហៅថា if-else ដែលមានរូបមន្តប្រើប្រាស់ដូចខាងក្រោម។

```
if ( លក្ខខណ្ឌពិត ) {
  // ដំណើរការកូដទាំងនេះ
} else {
  // ដំណើរការកូដទាំងនេះជំនួសវិញ
}
```

ដូចយើងបានឃើញ កូដនៅជួរបន្ទាប់ពី if នឹងដំណើរការប្រសិនបើលក្ខខណ្ឌដែលតេស្តចេញលទ្ធផលពិត។ ផ្ទុយពីនេះ វានឹងដំណើរការកូដជួរបន្ទាប់ពី else។ if-else អាចឲ្យយើង សរសេរកូដងាយជាងមុនបាន ដូចបង្ហាញខាងក្រោម។

```
void loop( ) {
    if ( counter % 2 == 1 ) {
      digitalWrite( led1, LOW ); // turn the LED off by making the voltage LOW
      digitalWrite( led2, HIGH ); // turn the LED on ( HIGH is the voltage level )
    } else {
      digitalWrite( led1, HIGH ); // turn the LED on ( HIGH is the voltage level )
      digitalWrite( led2, LOW ); // turn the LED off by making the voltage LOW
    }
    delay( 1000 ); // wait for a second
    counter = counter + 1;
}
```

សំគាល់ផងដែរថា យើងអាចលុបបំបាត់ delay( ) មួយ ដោយប្រើប្រាស់ if-else ហើយធ្វើតេស្តលក្ខខណ្ឌ counter % 2 តែម្តងដែរ។ យើងអាចធ្វើឲ្យវាកាន់តែសាមញ្ញបន្ថែមទៀត ដោយប្រើប្រាស់កូដខាងក្រោម។

```
void loop( ) {
    if ( counter % 2 == 1 ) {
```

```
    led1 = 13;
    led2 = 12;
  } else {
    led1 = 12;
    led2 = 13;
  }
  digitalWrite( led1, HIGH ); // turn the LED on ( HIGH is the voltage level )
  digitalWrite( led2, LOW ); // turn the LED on ( HIGH is the voltage level )
  delay( 1000 ); // wait for a second
  counter = counter + 1;
}
```

ក្នុងករណីនេះ យើងអាចប្តូរដើម្បីបញ្ច្រាស pin របស់ LED អាស្រ័យលើលក្ខខណ្ឌតេស្តរបស់ if បន្ទាប់មកដំណើរការ function digitalWrite( )។ កម្មវិធីថ្មីនេះ ធ្វើការងារដូចកម្មវិធីមុនដែរ ហេតុនេះមានចំណុចមួយចំនួនដែលយើងអាចរៀនសូត្របាន។ មេរៀនទី១ ការប្រើ if សាមញ្ញ ល្មមគ្រប់គ្រាន់ធ្វើឲ្យកម្មវិធីដំណើរការ ប៉ុន្តែ if-else ធ្វើឲ្យវាដំណើរការកាន់តែមានប្រសិទ្ធភាព។ មេរៀនទី២ ចំណែកឯ if-else ទី២ ធ្វើឲ្យកាន់តែងាយក្នុងការអាន និងយល់។ មេរៀនទី២នេះ អាចឲ្យយើងរៀនបានមេរៀនទី៣បន្ថែមទៀត គឺមានរបៀបច្រើនក្នុងការសរសេរកម្មវិធីឲ្យធ្វើការដូចគ្នា។ គ្រាន់តែសរសេរកម្មវិធីដែលអាចដំណើរការបាន មិនមានន័យថាវាជាកម្មវិធីដែលល្អ និងមានប្រសិទ្ធភាពទេ។ នៅពេលយើងប្រើប្រាស់ microcontroller ដែលមាន memory តូច ការកែសម្រួលកូដបន្តិចបន្តួចអាចជួយឲ្យកម្មវិធីដំណើរការបាន ឬ អាចធ្វើឲ្យកម្មវិធីប្រើប្រាស់ memory ហួសកម្រិត។

ការប្រើប្រាស់ if ច្រើនស្រទាប់

ជាញឹកញយ កម្មវិធីតម្រូវឲ្យធ្វើសកម្មភាពផ្សេងៗ អាស្រ័យទៅតាមតម្លៃរបស់អញ្ញត្តិ។ ឧទាហរណ៍ គេអាចមានអញ្ញត្តិមានឈ្មោះថា myDay ដែលមានតម្លៃ 1 (Sunday) រហូតដល់ 7 (Saturday) ដែលមានកូដដូចខាងក្រោម។

```
int myDay;
// Some code that determines what day it is...
if ( myDay == 1 ) {
        doSundayStuff( );
```

```
}
if ( myDay == 2 ) {
        doMondayStuff( );
      }
if ( myDay ==3 ) {
        doTuesdayStuff( );
}
if ( myDay == 4 ) {
        doWednesdayStuff( );
}
if ( myDay == 5 ) {
        doThursdayStuff( );
}
if ( myDay == 6 ) {
        doFridayStuff( );
}
if ( myDay == 7 ) {
        doSaturdayStuff( );
      }
```

កូដបែបនេះមិនមែនជាការសរសេរកូដដែលល្អទេ ដោយសារវាដំណើរការកូដដែលមិនចាំបាច់ជាច្រើន។ ឧទាហរណ៍ ប្រសិនបើ myDay ស្មើនឹង 1 (Sunday) ការធ្វើតេស្តដោយប្រើប្រាស់ if ទី១ នឹងផ្តល់លទ្ធផលពិត វានឹងដំណើរការ function doSundayStuff( )។ បញ្ហាគឺវានឹងបន្តធ្វើតេស្តមិនចាំបាច់ប្រាំមួយដងទៀត ទោះបីជាយើងដឹងថាវានឹងផ្តល់លទ្ធផលមិនពិតក៏ដោយ។ ដើម្បីដោះស្រាយបញ្ហានេះ C អនុញ្ញាតឲ្យយើងប្រើប្រាស់ if នៅក្នុង if ដូចបង្ហាញក្នុងឧទាហរណ៍ខាងក្រោម។

```
if ( myDay == 1 ) {
        doSundayStuff( );
} else {
        if ( myDay == 2 ) {
          doMondayStuff( );
```

```
  } else {
    if ( myDay == 3 ) {
      doTuesdayStuff( );
    } else {
      // you get the idea...
    }
  }
}
```

នៅពេល myDay ស្មើនឹង 1 វាដំណើរការ doSundayStuff( ) ហើយការធ្វើតេស្តដែលនៅសល់ត្រូវរំលង ដោយសារ else ទីមួយនឹងមិនដំណើរការទេ ប្រសិនបើការធ្វើតេស្តនៅ if ទីមួយផ្តល់លទ្ធផលពិត។

សញ្ញាបូកបន្ថែម ឬ ដកមួយ

នៅក្នុងកូដកម្មវិធីកន្លងមករបស់យើង បន្ទាត់ចុងក្រោយគឺ counter = counter + 1 ដែលបូកបន្ថែមតម្លៃ 1 ទៅឲ្យអញ្ញត្តិ counter។ ដោយសារគេឧស្សាហ៍ប្រើប្រាស់វា C មានបង្កើតសញ្ញាបូកបន្ថែមសម្រាប់ការងារនេះគឺ ++។

វាអាចមានពីរប្រភេទគឺ បូកបន្ថែមមុន (ដាក់ខាងមុខអញ្ញត្តិ) និងបូកបន្ថែមក្រោយ (ដាក់នៅខាងក្រោយអញ្ញត្តិ)។

```
++counter; // pre-increment
```

បានន័យថា counter ត្រូវបានបូកបន្ថែមតម្លៃ 1 ជាមុនសិនមុននឹងប្រើប្រាស់។ ចំណែកឯ

```
counter++; // post-increment
```

តម្លៃអញ្ញត្តិប្រើប្រាស់ជាមុនសិន មុននឹងបូកបន្ថែម 1។

យើងអាចឆ្ងល់ តើមានភាពខុសគ្នាយ៉ាងម៉េច? ឧទាហរណ៍កូដខាងក្រោម

```
int c = 5;
```

```
int k;
    k = ++c; // pre-increment, k == 6, c == 6
```

ក្នុងករណីនេះ តម្លៃ k ស្មើនឹង 6 ហើយតម្លៃ c ស្មើនឹង 6 ដែរ។ ដោយសារយើងដាក់ ++ ខាងមុខ c។ ចំណែកឯ កូដខាងក្រោមនេះវិញ

```
k = c++; // post-increment, k == 5, c == 6
```

តម្លៃ k ស្មើនឹង 5 ចំណែកឯ c ស្មើនឹង 6។

ចំណែកឯសញ្ញាដកតម្លៃ 1 (--) ក៏ស្រដៀងទៅនឹងសញ្ញាបូកបន្ថែមតម្លៃ 1 ដែរ។

```
counter = counter – 1;
```

អាចសរសេរ

```
counter--;
```

ការប្រើប្រាស់ switch

switch ប្រើប្រាស់រូបមន្តដូចខាងក្រោម

```
switch (expression1) { // opening brace for switch statement block
      case 1:
        // statements to execute when expression1 is 1
        break;
      case 2:
        // statements to execute when expression1 is 2
        break;
      case 3:
        // statements to execute when expression1 is 3
        break;
```

```
    // more case statements as needed
    default:
    // statements to execute if expression1 doesn't have a "case value"
      break;
} // close brace for switch statement block
  // This is the next statement after the switch
```

ប្រើប្រាស់ឧទាហរណ៍អញ្ញត្តិ myDay ពី if ច្រើនស្រទាប់កន្លងមក case នីមួយៗត្រូវនឹងថ្ងៃនៃសប្តាហ៍។ ប្លុក default សម្រាប់ករណី myDay ស្មើនឹង 7។ ប្រសិនបើ expression1 មានតម្លៃខុសពី 1 ទៅ 7 កូដនៅក្នុងប្លុក default នឹងដំណើរការ។ expression1 អាចជាប្រភេទ byte, char, int ឬ long (រួមទាំង unsigned ផងដែរ) មិនអាចជាប្រភេទ float ឬ double បានទេ។

៣.២.៥ ការធ្វើឲ្យកម្មវិធីដំណើរការសារចុះឡើង

ការងារមួយដែលកុំព្យូទ័រធ្វើការងារបានល្អជាងមនុស្សគឺការធ្វើការងារដដែលៗ។ មនុស្សអាចធុញទ្រាន់ ពេលធ្វើការងារដដែលៗ បណ្តាលឲ្យគេមិនសូវយកចិត្តទុកដាក់ និងបង្កើតកំហុស។ ចំណែកកុំព្យូទ័រវិញ មិនចេះធុនទ្រាន់ទេ ហេតុនេះវាធ្វើការងារដដែលៗបានល្អ។ Microcontroller នឹងនៅតែបន្តដំណើរការ ដដែលៗ រហូតដល់ដាច់ចរន្តអគ្គីសនី ឬ ផ្នែកណាមួយរបស់វាខូច ឬ យើងបញ្ជាឲ្យវាធ្វើអ្វីផ្សេង។

ការធ្វើឲ្យកម្មវិធីដំណើរការសារចុះឡើងប្រើប្រាស់ for

រូបមន្តទូទៅសម្រាប់ប្រើប្រាស់មានដូចខាងក្រោម

```
  for ( expression1; expression2; expression3 ) {
    // for loop statement body
}
  // the first statement following the for loop structure
```

វាមានពាក្យគន្លឹះ for បន្ទាប់មកមានសញ្ញា (។ បន្តមកទៀតមាន expression បី ផ្តាច់ពីគ្នាដោយសញ្ញា ; ។ expression3 បញ្ចប់ដោយសញ្ញា)។ នៅចន្លោះ {……} ជាកូដកម្មវិធីដែលត្រូវដំណើរការសារចុះឡើង។ expression1 ជាតម្លៃចាប់ផ្តើមបញ្ជាដំណើរការធ្វើសារចុះឡើង។ expression2 តេស្តដើម្បីដឹងថាត្រូវ

បន្តទៀត ឬ ឈប់ត្រឹមនេះ។ exression3 ផ្លាស់ប្តូរភាពនៃអញ្ញត្តិដែលបញ្ជាដំណើរការធ្វើសារចុះឡើង។ ខាងក្រោមនេះជាឧទាហរណ៍នៃការប្រើប្រាស់ for។

```
int k;

for( k = 0; k < 1000; k++ ) {
  DoSomethingCool( k );
}
```

កម្មវិធីនេះចាប់ផ្តើមដោយតាងអញ្ញត្តិ k ជាប្រភេទលេខគត់ បន្ទាប់មកទៀតគឺជាប្លុក for ។ exression1 (k = 0) បញ្ចប់ដោយសញ្ញា ; បានន័យថាវាបានបញ្ចប់ ចូរសំគាល់ថា expression1 អាចមាន expression វែង ផ្តាច់ពីគ្នាដោយសញ្ញា , ដូចក្នុងឧទាហរណ៍ខាងក្រោម។

```
for ( k = 0, j = 1; k < 1000; k++ ) {
```

ដែល j ស្មើនឹង 1 ជាផ្នែកនៃ expression1។

នៅពេល expression1 វាយតម្លៃរួចហើយ វាបន្តទៅវាយតម្លៃ expression2 ទៀត (k < 1000)។ ប្រសិនបើ វាមានលទ្ធផល true វានឹងបន្តទៀតកូដក្នុង { } រួចវាចាប់ផ្តើមវាយតម្លៃ expression3 ដែលជា k++ បានន័យថាបូកបន្ថែមតម្លៃ 1 ពីលើតម្លៃ k បច្ចុប្បន្ន។ ប្រសិនបើ expression2 វាយតម្លៃចេញលទ្ធផល false វានឹងបញ្ចប់ដំណើរការសារចុះឡើងរបស់ for ហើយបន្តដំណើរការកូដនៅខាងក្រោមប្លុក for បន្ទាប់ពីសញ្ញា }។ expression3 អាចមាន expression វែង ដូចបង្ហាញខាងក្រោមនេះដែរ។

```
for ( k = 0; k < 1000; k++, j-- ) {
```

គេប្រើប្រាស់ for នៅពេលដឹងចំនួនសារចុះឡើងដែលត្រូវធ្វើ។

ការធ្វើឲ្យកម្មវិធីដំណើរការសារចុះឡើងប្រើប្រាស់ while

ខាងក្រោមនេះជារូបមន្តសម្រាប់ប្រើប្រាស់ while

```
while ( expression2 ) {
  // Statements in the loop body
```

```
} // End of while statement block
```

សូមសំគាល់ថា មានតែ expression2 ទេ ដែលត្រូវវាយតម្លៃដើម្បីដឹងថាត្រូវបន្តធ្វើសារចុះឡើង ឬឈប់។ គេអាចបំលែង for ទៅជា while ដោយងាយស្រួល។ ក្នុងកូដខាងក្រោមនេះ ឧទាហរណ៍ការធ្វើសារចុះឡើងប្រើប្រាស់ for ពីមុនត្រូវបានកែសម្រួលមកប្រើប្រាស់ while វិញ។

```
int k;
// some additional statements
k = 0; // This is expression1
    while ( k < 1000 ) {
      // This is expression2
      DoSomethingCool( );
      k++;
      // This is expression3
    } // End of the while loop
```

ការធ្វើឲ្យកម្មវិធីដំណើរការសារចុះឡើងប្រើប្រាស់ do-while

រូបមន្តសម្រាប់ប្រើប្រាស់ do-while ដូចបង្ហាញខាងក្រោម

```
do {
      // Loop body statements
    } while ( expression2 );
```

do-while ខុសពី while ត្រង់កូដក្នុងប្លុក { ………… } ត្រូវដំណើរការយ៉ាងតិចម្តងមុននឹងឈប់។ សូមពិនិត្យមើលកូដខាងក្រោម។

```
int k = 1001;
while ( k < 1000 ) {
      DoSomethingCool( k );
      k++;
}
```

ពេលចាប់ផ្តើមដំណើរការដំបូង ការតេស្ត k < 1000 បរាជ័យ ដោយសារតម្លៃរបស់ k ស្មើនឹង 1001។ ហេតុនេះ function DoSomethingCool( ) មិនអាចដំណើរការបានទាល់តែសោះ។ ឥឡូវនេះ យើងកែសម្រួលវា ប្រើប្រាស់ do-while វិញ ដូចបង្ហាញខាងក្រោម។

```
int k = 1001;
do {DoSomethingCool( k );
        k++;
} while ( k < 1000 );
```

ក្នុងករណីនេះ function DoSomethingCool( ) ដំណើរការបានម្តងទោះបីតម្លៃ k = 1001 ក៏ ដោយ។

ពាក្យគន្លឹះ break និង continue

គេច្រើនឃើញប្រើប្រាស់ពាក្យទាំងពីរនេះក្នុងការធ្វើសារចុះឡើង។ ឧទាហរណ៍ពីរខាងក្រោមនេះបង្ហាញពីការប្រើប្រាស់ break និង continue។

ឧទាហរណ៍ការប្រើប្រាស់ break

```
#define MAXVATCOUNT 200
#define GOALTEMPERATURE 160
// Some statements...
int vatTemperature;
int counter = 0;
loop( ) {
        while ( counter < MAXVATCOUNT ) {
          vatTemperature = ReadVatTemp( counter );
          if ( vatTemperature == GOALTEMPERATURE ) {
            break;
          }
          counter++;
```

```
    if ( counter == MAXVATCOUNT )
      counter = 0;
  }
  AddChemicals( counter );
  if ( counter < MAXVATCOUNT ) {
    counter++;
  } else {
    counter = 0; // Just in case this is the last vat
  }
}
```

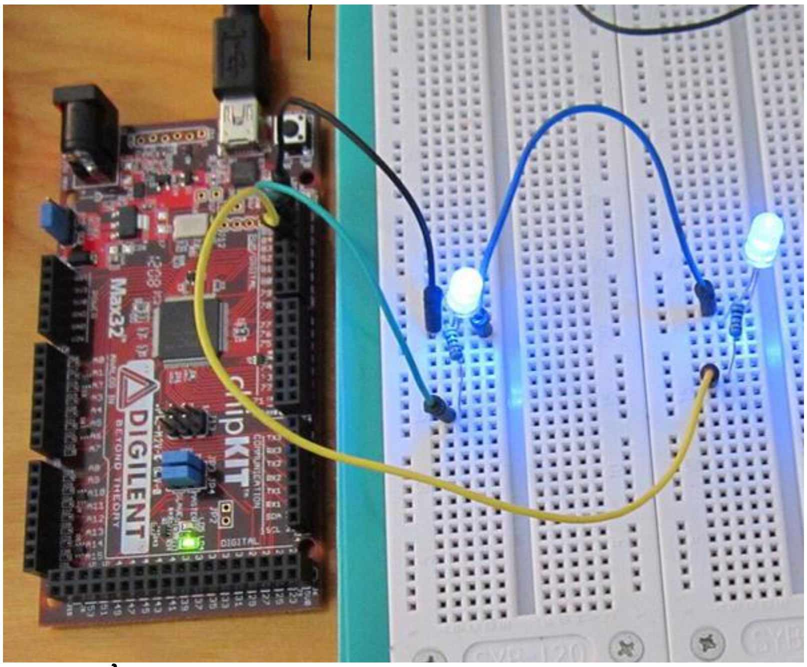
ឧទាហរណ៍ការប្រើប្រាស់ continue

```
#define MAXVATCOUNT 200
#define GOALTEMPERATURE 160
// Some statements...
int vatTemperature;
int counter = 0;
loop( ) {
  while ( counter <= MAXVATCOUNT ) {
    vatTemperature = ReadVatTemp( counter );
    if ( vatTemperature != GOALTEMPERATURE ) { // Big
difference here...
      counter++;
      if ( counter > MAXVATCOUNT )
        counter = 0;
        continue;
      }
    AddChemicals( counter );
    if ( counter < MAXVATCOUNT ) {
      counter++;
```

```
                    } else {
                      counter = 0; // Just in case this is the last vat
                    }
                  }
}
```

ឧទាហរណ៍កូដទាំងស្រុង

យើងនឹងប្រើប្រាស់សៀគ្វីធ្វើឲ្យអំពូល LED ភ្លឺភ្លឹបភ្លែតកន្លងមកឡើងវិញ។ ប៉ុន្តែនៅពេលនេះ យើងនឹងប្រើប្រាស់ការបង្កើតលេខចៃដន្យ ហើយរកតម្លៃជាក់លាក់ដើម្បីផលិត។ នៅពេលរកឃើញតម្លៃលេខជាក់លាក់ដែលចង់បាន វានឹងធ្វើឲ្យ LED តំណាងឲ្យការរកឃើញភ្លឺ ១ វិនាទី ហើយផ្ញើសារទៅ Serial Monitor នឹងបង្ហាញតម្លៃ counter។ នៅពេលស្វែងរករាល់ជុំ LED មួយទៀតនឹងភ្លឺរយៈពេល១វិនាទី ផ្ញើសារទៅ Serial Monitor ដើម្បីបង្ហាញចំនួនវិលជុំ។ យើងនឹងកំណត់លេខចៃដន្យនៅចន្លោះ 0 ទៅ 5000 ដើម្បីកុំឲ្យចំណាយពេលស្វែងរកយូរពេក។



រូបភាព ៣.៣៥ ៖ សៀគ្វីធ្វើតេស្តការផ្ញើសារចុះឡើង

ដំណើរការសរសេរកូដមាន៥ជំហ៊ាន ៖

ជំហ៊ានទី១ ការចាប់ផ្តើមដំបូង

ត្រូវកំណត់ pin ដែលតភ្ជាប់ទៅ LED ល្បឿនបញ្ជូនទិន្នន័យទៅ Serial Monitor បង្កើតតម្លៃគោលដៅកំណត់អញ្ញត្តិផ្សេងៗ និងតម្លៃលេខចៃដន្យ។

ជំហានទី២ ធាតុចូល

ជាតម្លៃលេខចៃដន្យបង្កើតដោយ microcontroller។

ជំហានទី៣ កែច្នៃ

យើងត្រូវពិនិត្យមើលតើលេខចៃដន្យស្មើនឹងតម្លៃគោលដៅរបស់យើងដែរឬទេ? ប្រសិនបើត្រូវ ឲ្យ LED តំណាងឲ្យការរកឃើញភ្លឺ។ យើងក៏បង្កើនតម្លៃអញ្ញត្តិ counter ហើយកំណត់ថាវានៅតែវិជ្ជមាន ឬ អត់? ប្រសិនបើអវិជ្ជមាន ត្រូវឲ្យ LED តំណាងឲ្យរកមួយជុំទៀតភ្លឺ។

ជំហានទី៤ ធាតុចេញ

ប្រសិនបើរកឃើញលេខចៃដន្យ ត្រូវបើក LED តំណាងឲ្យការរកឃើញភ្លឺរយៈពេល១វិនាទី។ យើងត្រូវផ្ញើសារទៅកាន់កុំព្យូទ័រ (Serial Monitor) តម្លៃ counter បច្ចុប្បន្ន។ ប្រសិនបើ អញ្ញត្តិ counter អវិជ្ជមាន ត្រូវបើកភ្លើងរកមួយជុំទៀតរយៈពេល១វិនាទី និងផ្ញើសារទៅកុំព្យូទ័រ។

ជំហានទី៥ ការបញ្ចប់

ប្រសិនបើ LED តំណាងឲ្យការរកម្តងទៀតភ្លឺប្រាំដង ត្រូវបញ្ចប់កម្មវិធី។ Serial Monitor នឹងប្រាប់យើងនៅពេលដែលកម្មវិធីបញ្ចប់។

ខាងក្រោមនេះជាកូដទាំងអស់។

```
// define the pins to be used.
#define MAX 5000L
#define MIN 0L
#define TARGETVALUE 2500L
#define MAXRECYCLES 5
#define FOUNDITIOPIN 13
#define RECYCLEIOPIN 12
#define PAUSE 1000
int foundIt = FOUNDITIOPIN;
int recycle = RECYCLEIOPIN;
long targetValue = TARGETVALUE;
long randomNumber;
int recycleCounter = 0;
int counter = 0;
// the setup routine runs once when you press reset:
```

```
void setup( ) {
  // initialize each of the digital pins as an output.
  Serial.begin( 9600 );
  pinMode( foundIt, OUTPUT );
  pinMode( recycle, OUTPUT );
  randomSeed( analogRead( 0 ) ); // This seeds the random
number generator
}
// the loop routine runs over and over again forever:
void loop( ) {
  while ( counter != -1 ) { // Check for negative values
    randomNumber = generateRandomNumber( );
    if ( randomNumber == TARGETVALUE ) {
      Serial.print( "Counter = " );
      Serial.print( counter, DEC );
      Serial.print( " recycleCounter = " );
      Serial.println( recycleCounter, DEC );
      digitalWrite( foundIt, HIGH );
      delay( PAUSE );
      digitalWrite( foundIt, LOW );
    }
    counter++;
    if ( counter < 0 ) { // We've overflowed an int
      counter = 0;
      recycleCounter++;
      Serial.print( "recycleCounter = " );
      Serial.println( recycleCounter, DEC );
      digitalWrite( recycle, HIGH );
      delay( PAUSE );
      digitalWrite( recycle, LOW );
    }
```

```
        if ( recycleCounter == MAXRECYCLES )
          exit( 0 ); // End program
        }
      }
      long generateRandomNumber( )
      {
        return random( MIN, MAX ); // Generate random numbers
between 0 and one million
        }
```

កូដខាងលើធ្វើការដូចអ្វីដែលយើងចង់បាន។ នៅពេលដែលមើលឲ្យល្អិតល្អន់ ហើយសួរខ្លួនឯងថា តើការធ្វើតេស្តដោយ while មានឱកាសឃើញតម្លៃអវិជ្ជមានដែរឬទេ? ចម្លើយគឺមិនអាចទៅរួចទេ ដោយសារវាប្តូរតម្លៃទៅស្មើនឹង 0 វិញ ប្រសិនបើតម្លៃរបស់វាអវិជ្ជមាន។ ដូច្នេះយើងអាចជំនួសការធ្វើតេស្ត while ដោយកូដខាងក្រោមវិញ។

```
while ( true ) {
```

វាធ្វើសារចុះឡើងមិនកំណត់ ហើយអាចបញ្ចប់កម្មវិធីដោយប្រើប្រាស់ function exit( )។

នៅពេលយើងពិនិត្យឃើញកូដសរសេរដដែលៗ យើងអាចកែសម្រួលវាឲ្យសាមញ្ញ។ ឧទាហរណ៍ កូដខាងក្រោម

```
digitalWrite( foundIt, HIGH );
delay( PAUSE );
digitalWrite( foundIt, LOW );
anddigitalWrite( recycle, HIGH );
delay( PAUSE );
digitalWrite( recycle, LOW );
```

ស្ទើរតែដូចគ្នា ហេតុនេះអាចជំនួសដោយ

```
ToggleLED( foundIt, PAUSE );
```

និង

```
ToggleLED( recycle, PAUSE );
```

រួចសរសេរ function ថ្មីដូចខាងក្រោម

```
void ToggleLED( int whichLED, int howLong ) {
  digitalWrite( whichLED, HIGH );
  delay( howLong );
  digitalWrite( whichLED, LOW );
}
```

៣.២.៦ Function នៅក្នុងភាសាកុំព្យូទ័រ C

Function ជាតួនៃកូដដែលបង្កើតឡើងសម្រាប់ធ្វើការងារណាមួយ។ មាន function រាប់រយដែលគេអាចប្រើប្រាស់បានពី function library ជាច្រើន ដូចជា function Math និង Time ដែលគេបានសរសេររួចហើយ។

រចនាសម្ព័ន្ធនៃ function របស់ភាសាកុំព្យូទ័រ C មានបង្ហាញដូចខាងក្រោម ដើម្បីតេស្តមើលថាតើជាឆ្នាំបង្គ្រប់ ឬ មិនមែន។

```
int IsLeapYear( int year )
{
  // The statements that perform the task Function body
}
```

int ជា function type specifier ដែលកំណត់ប្រភេទទិន្នន័យដែល function បញ្ចេញ ក្នុងករណីនេះជាលេខគត់ (int)។ ប្រសិនបើ function មិនបញ្ចេញតម្លៃគេប្រើប្រាស់ប្រភេទទិន្នន័យ void ជំនួសវិញ។

IsLeapYear ជាឈ្មោះរបស់ function ដើម្បីជួយក្នុងការសំគាល់។ ការដាក់ឈ្មោះគួរតែអាចឲ្យងាយដឹងពីមុខងាររបស់វា។

int year ជា function argument(s) គេប្រើប្រាស់វាដើម្បីផ្តល់ទិន្នន័យទៅឲ្យ function ក្នុងឧទាហរណ៍របស់យើង វាជាលេខឆ្នាំដែលយើងចង់ដឹងថាជាឆ្នាំបង្គ្រប់ ឬ មិនមែន ?

Function Body ចាប់ផ្តើមដោយសញ្ញា { និង បញ្ចប់ដោយសញ្ញា }។ ប្រសិនបើ function type specifier ខុសពី void គេត្រូវប្រើប្រាស់ពាក្យគន្លឹះ return យ៉ាងតិច១ នៅក្នុង function body ជាពិសេសនៅមុនបញ្ចប់ function body ដូចបង្ហាញក្នុងកូដខាងក្រោម។

```
int VolumeOfCube( int width, int length, int height )
{
  int volume;
  volume = width  * length  * height;
  return volume;
}
```

អ្នកសរសេរកម្មវិធីមានបទពិសោធន៍ច្រើន ច្រើនសរសេរកូដឲ្យខ្លីតាមដែលអាចធ្វើទៅបាន។ គេអាចសរសេរកូដខាងលើនេះ មកជា

```
int VolumeOfCube( int width, int length, int height )
{
  return width  * length  * height;
}
```

ការសរសេរ function ដោយខ្លួនឯង

គេសរសេរ function ដោយខ្លួនឯងនៅពេលមិនអាចរក function ដែលមានស្រាប់នៅក្នុង function library។ Function មួយចំនួនធំ មានគេសរសេររួចរាល់ហើយ ហេតុនេះយើងមិនចាំបាច់ចំណាយពេលឥតប្រយោជន៍ដើម្បីសរសេរ function ខ្លួនឯងទេ។ ឧទាហរណ៍ function IsLeapYear( ) មាននៅក្នុង standard library រួចហើយ ប៉ុន្តែយើងអាចសរសេរ function នេះដោយខ្លួនឯងបានផងដែរ។ ខាងក្រោមនេះ ជាកូដប្រើប្រាស់ standard library។

```
int daysInFeb;
if ( IsLeapYear( year ) == true )
  daysInFeb = 29;
else
  daysInFeb = 28;
```

កូដនេះមិនសូវជាល្អទេ ហេតុនេះយើងត្រូវតាងតម្លៃអញ្ញត្តិ daysInFeb និងលុប else ចោលដូចខាងក្រោម

```
int daysInFeb = 28;
if ( IsLeapYear( year ) == true )
  daysInFeb = 29;
```

Function ពី standard library នេះ ត្រូវការប្រើ if ដើម្បីប្តូរចំនួនថ្ងៃនៅក្នុងខែកុម្ភៈក្នុងករណីជាឆ្នាំបង្គ្រប់។ ប្រសិនបើយើងសរសេរ function ថ្មីដោយខ្លួនឯងដែលបញ្ចេញតម្លៃគត់ស្មើនឹង 1 នៅពេលជាឆ្នាំបង្គ្រប់ និងស្មើនឹង 0 ក្នុងករណីមិនមែន ពេលនោះយើងអាចសរសេរកូដដូចខាងក្រោម។

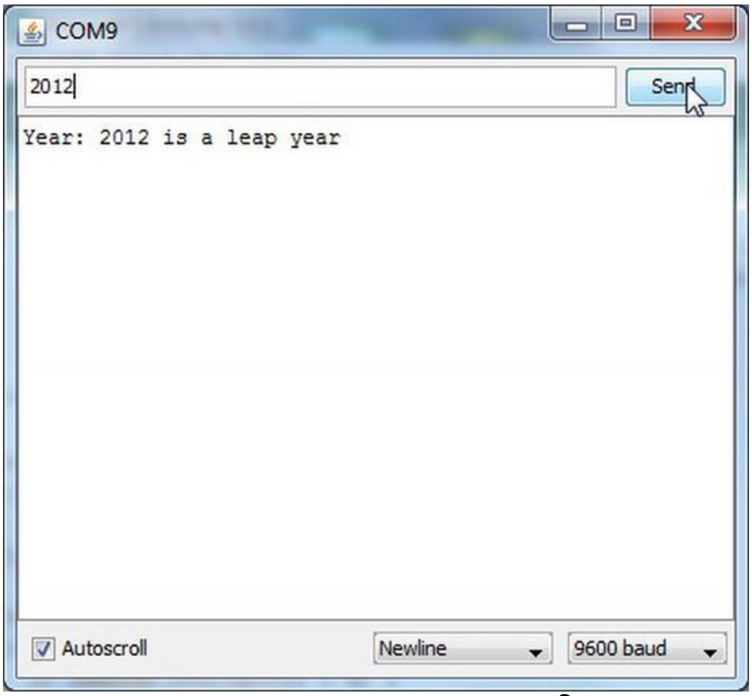
```
int daysInFeb = 28 + IsLeapYear( year );
```

ឧទាហរណ៍ ខាងក្រោមនេះជាកូដកម្មវិធីទាំងស្រុង ដែលទទួលលេខឆ្នាំពីអ្នកប្រើប្រាស់កុំព្យូទ័រតាម រយៈ Serial Monitor ហើយកំណត់ថាវាជាឆ្នាំបង្គ្រប់ ឬ មិនមែន ដោយវាយលេខឆ្នាំក្នុង textbox រួចចុចប៊ូតុង Send ដូចបង្ហាញក្នុងរូបភាព ៣.៣៦។ កម្មវិធីនេះមានប្រើប្រាស់ចំណេះដឹងដែលទើបបានពិភាក្សាកន្លងមកនេះ ជាពិសេសការសរសេរ function ដោយខ្លួនឯង។

```
/* *
Program: find out is the user typed in a leap year. The code assumes
the user is not an idiot and only types in numbers that are a valid
year.
Author: Dr. Purdum, Aug. 7, 2012
 * */
void setup( )
{
  Serial.begin( 9600 );
}
void loop( )
{
```

```
  if ( Serial.available( ) > 0 ) {
    int bufferCount;
    int year;
    char myData[20];
    bufferCount = ReadLine( myData );
    year = atoi( myData ); // Convert to int
    Serial.print( "Year: " );
    Serial.print( year );
    if ( IsLeapYear( year ) ) {
      Serial.print( " is " );
    } else {
      Serial.print( " is not " );
    }
    Serial.println( "a leap year" );
  }
}

/* * * * *
Purpose: Determine if a given year is a leap year
Parameters:
int yr The year to test
Return value:
int 1 if the year is a leap year, 0 otherwise
 * * * * */
int IsLeapYear( int yr )
{
  if ( yr % 4 == 0 && yr % 100 != 0 || yr % 400 == 0 ) {
    return 1; // It is a leap year
} else {
    return 0; // not a leap year
  }
```

```
}

/*****
Purpose: Read data from serial port until a newline character is read ('\n')
Parameters:
char str[] character array that will be treated as a nul-terminated string
Return value:
int the number of characters read for the string
CAUTION: This method will sit here forever if no input is read from the serial
port and no newline character is entered.
****/
int ReadLine(char str[])
{
  char c;
  int index = 0;
  while (true) {
    if (Serial.available() > 0) {
      c = Serial.read();
      if (c != '\n') {
        str[index++] = c;
      } else {
        str[index] = '\0'; // null termination character
        break;
      }
    }
  }
  return index;
}
```



រូបភាព ៣.៣៦ ៖ Serial Monitor បង្ហាញពីការប្រើប្រាស់កម្មវិធីតេស្តឆ្នាំបង្គ្រប់

៣.៣ សេនស័រសីតុណ្ហភាពនិងសំណើមបរិយាកាស

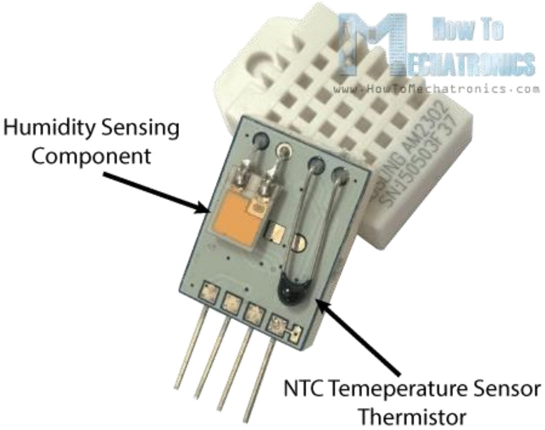
សេនស័រសីតុណ្ហភាពនិងសំណើមបរិយាកាសដែលគេនិយមប្រើប្រាស់ជាមួយ Arduino គឺប្រភេទ DHT11 និង DHT22 ដូចបង្ហាញក្នុងរូបភាព ៣.៣៧។ DHT22 មានភាពសុក្រឹតជាង DHT11 ហើយចន្លោះសីតុណ្ហភាពនិងសំណើមបរិយាកាសដែលវាអាចវាស់បានក៏ធំជាងដែរ។ ចំណែកឯល្បឿនវាស់ (sampling rate) ក៏លឿនជាងផងដែរ។

0 - 50°C / ± 2°C	*Temperature Range*	-40 - 125 °C / ± 0.5 °C
20 - 80% / ± 5%	*Humidity Range*	0 - 100 % / ± 2-5%
1Hz (one reading every second)	*Sampling Rate*	0.5 Hz (one reading every two seconds)
15.5mm x 12mm x 5.5mm	*Body Size*	15.1mm x 25mm x 7.7mm
3 - 5V	*Operating Voltage*	3 - 5V
2.5mA	*Max Current During Measuring*	2.5mA

រូបភាព ៣.៣៧ ៖ លក្ខណៈបច្ចេកទេសរបស់ DHT11 និង DHT22

ប្រភព ៖ https://www.howtomechatronics.com

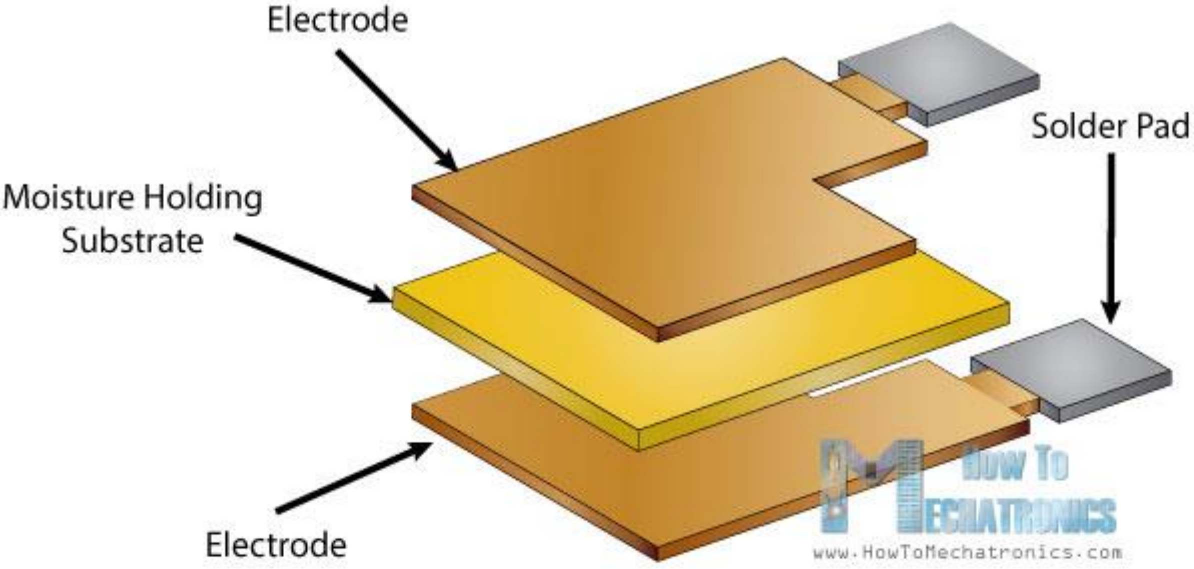
គោលការណ៍ដំណើរការរបស់វាប្រើ Humidity Sensing Component សម្រាប់វាស់សំណើមបរិយាកាស និងប្រើ Thermistor សម្រាប់វាស់សីតុណ្ហភាព ដូចបង្ហាញក្នុងរូបភាព ៣.៣៨។



រូបភាព ៣.៣៨ ៖ គោលការណ៍ដំណើរការរបស់សេនស័រ DHT

ប្រភព ៖ https://www.howtomechatronics.com

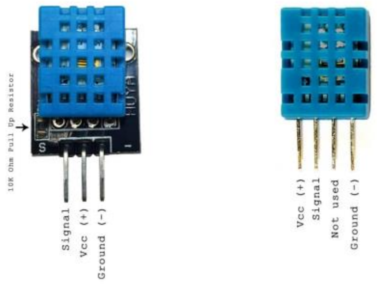
ជាតុផ្សំរបស់ Moisture Sensing Component មានបង្ហាញក្នុងរូបភាព ៣.៣៩ ដែលមានអេឡិកត្រូតពីរមានដាក់សារធាតុស្រូបសំណើមនៅចន្លោះនោះ។ នៅពេលសំណើមប្រែប្រួល ធ្វើឲ្យកម្រិតចម្លងអគ្គីសនីរបស់វាប្រែប្រួលដែរ ដែលអាចឲ្យគេវាស់សំណើមបរិយាកាសបាន។



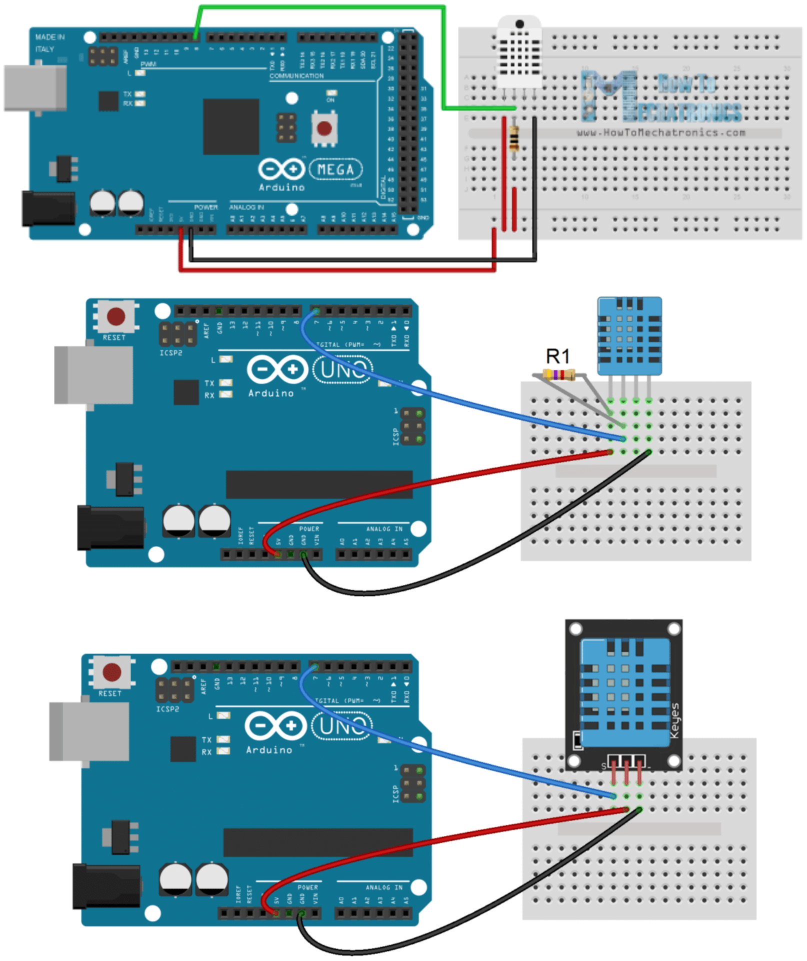
រូបភាព ៣.៣៩ ៖ ធាតុផ្សំរបស់ moisture sensing component របស់សេនស័រ DHT

ប្រភព ៖ https://www.howtomechatronics.com

របៀបតភ្ជាប់ទៅ Arduino ប្រែប្រួលទៅតាមក្រុមហ៊ុនដែលផលិតសូមពិនិត្យមើលរូបភាព ៣.៤០ និងរូបភាព ៣.៤១។ ជាធម្មតាមាន pin បី Vcc (+) តភ្ជាប់ទៅ pin ផ្តល់អគ្គីសនី 5V និង Ground (-) តភ្ជាប់ទៅ GND pin នៅលើ board របស់ Arduino។ អាស្រ័យទៅតាមក្រុមហ៊ុនផលិត Signal pin ខ្លះតភ្ជាប់ទៅនឹង digital pin ណាមួយរបស់ Arduino តែម្តង ខ្លះទៀតត្រូវតែបន្ថែម pin នេះតាមរយៈរ៉េស៊ីស្តង់ទៅ pin 5V ទៀត។ ហេតុនេះមុននឹងតភ្ជាប់ សូមពិនិត្យមើលតាមការណែនាំរបស់ក្រុមហ៊ុនផលិត។



រូបភាព ៣.៤០ ៖ Pin របស់សេនស័រ DHT



រូបភាព ៣.៤១ ៖ របៀបតភ្ជាប់សេនស័រ DHT ផ្សេងៗគ្នាទៅ Arduino

ប្រភព ៖ https://www.howtomechatronics.com

ឧទាហរណ៍កូដ

```
#include <dht.h>
dht DHT;
#define DHT11_PIN 11
void setup(){
  Serial.begin(9600);
}
void loop()
{
  int chk = DHT.read11(DHT11_PIN);
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);
  delay(1000);
}
```

រូបភាព ៣.៤២ ៖ កូដសម្រាប់បង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសពីសេនស័រ DHT11

ជួរខាងក្រោមនេះ

```
#include <dht.h>
```

ដើម្បីបន្ថែម dht library

```
dht DHT;
```

បង្កើតអញ្ញត្តិ DHT ជាប្រភេទទិន្នន័យ dht ដែលត្រូវការប្រើប្រាស់ function របស់វាដូចជា DHT.read11( ), DHT.temperature( ) និង DHT.humidity( ) នៅក្នុង loop( ) ខាងក្រោម។

កូដខាងក្រោមនេះ តាងអញ្ញត្តិ DHT11_PIN ដោយតម្លៃ 11 ដែល signal pin របស់ DHT តភ្ជាប់ទៅនឹង Arduino Uno។

```
#define DHT11_PIN 11
```

កូដខាងក្រោម នៅក្នុង setup( ) កំណត់ល្បឿនរបស់ serial port ដែល Arduino Uno តភ្ជាប់ទៅនឹងកុំព្យូទ័រ ហើយតម្លៃ 9600 ជាល្បឿនបញ្ជូនទិន្នន័យរបស់ serial port (baud rate)។ សម្រាប់ឧទាហរណ៍នេះ គេអាចប្រើប្រាស់ baud rate នេះបាន។ ករណីខ្លះត្រូវបញ្ជូនទិន្នន័យលឿន គេត្រូវបង្កើតតម្លៃ baud rate នេះ។

```
Serial.begin(9600);
```

ជួរខាងក្រោមនេះ តាងអញ្ញត្តិ chk ជាប្រភេទ int (លេខគត់) ដែលតម្លៃរបស់វាបានមកពី function DHT.read11(DHT11_PIN) ដែលអានតម្លៃពីសេនស័រ DHT11។

```
int chk = DHT.read11(DHT11_PIN);
```

កូដខាងក្រោមនេះ ដំបូងព្រីន Temperature = ទៅក្នុង serial monitor បន្ទាប់មកព្រីនតម្លៃសីតុណ្ហភាពដែលអានពី DHT11 បន្ទាប់មកចុះបន្ទាត់និងចាប់ផ្តើមបន្ទាត់ថ្មី ដោយសារប្រើប្រាស់ Serial.println() ជំនួសឲ្យ Serial.print()។ បន្តទៀត Arduino Uno នឹងព្រីន Humidity = និងតម្លៃសំណើមបរិយាកាស រួចចុះបន្ទាត់ និងចាប់ផ្តើមបន្ទាត់ថ្មី។ បន្តទៀត Arduino Uno នឹងរងចាំ ១ វិនាទីសិន មុននឹងវិលត្រលប់ទៅដំណើរការកូដជួរដំបូងនៅក្នុង function loop() ម្តងទៀត។

```
Serial.print("Temperature = ");
Serial.println(DHT.temperature);
Serial.print("Humidity = ");
Serial.println(DHT.humidity);
delay(1000);
```

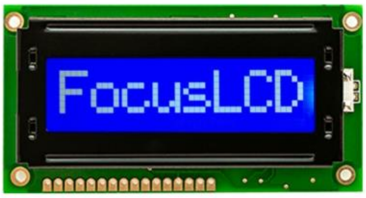
ហេតុនេះនៅពេលកូដនេះដំណើរការ វានឹងបង្ហាញសីតុណ្ហភាព និងសំណើមបរិយាកាសពីសេនស័រ DHT11 រៀងរាល់១វិនាទីម្តង។

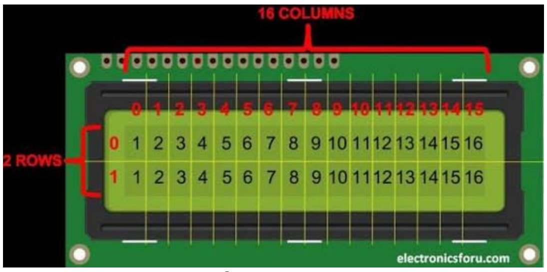
៣.៤ ការបង្ហាញព័ត៌មាននៅលើអេក្រង់ LCD

នៅក្នុងឧទាហរណ៍ បង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសពីសេនស័រ DHT11 កន្លងមកនេះ គេអាចមើលសីតុណ្ហភាពនិងសំណើមបរិយាកាសតាមរយៈ serial monitor នៅពេលតភ្ជាប់ Arduino Uno ទៅនឹងកុំព្យូទ័រ។ ករណីប្រើប្រាស់ដាច់តែឯង (មិនតភ្ជាប់ទៅនឹងកុំព្យូទ័រ) គេអាចប្រើប្រាស់អេក្រង់ LCD ដើម្បីបង្ហាញព័ត៌មានសីតុណ្ហភាពនិងសំណើមបរិយាកាសបាន។

៣.៤.១ ប្រភេទអេក្រង់ LCD

អេក្រង់ LCD មានច្រើនប្រភេទ ដូចជា 8x1, 16x2, 20x4 ដូចបង្ហាញក្នុងរូបភាព ៣.៤៣។





រូបភាព ៣.៤៣ ៖ ប្រភេទផ្សេងៗនៃអេក្រង់ LCD

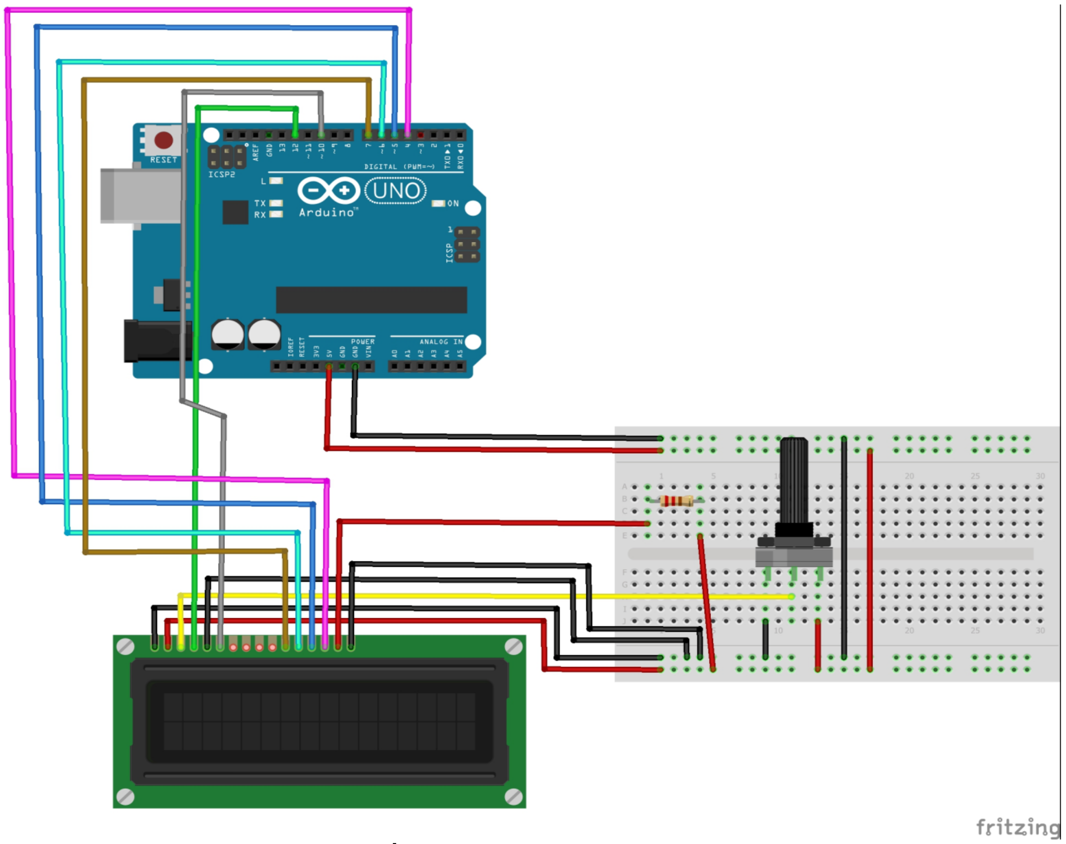
៣.៤.២ Pin របស់អេក្រង់ LCD

រូបភាព ៣.៤៤ ៖ Pin របស់អេក្រង់ LCD

Pin D0 ទៅ D7 គឺជាខ្សែបញ្ជូនទិន្នន័យ អាចប្រើខ្សែម្តង៨ ឬ ម្តង៤។ GND ជាខ្សែម៉ាស, VCC តទៅ +5V, Contrast អាចអោយភ្លឺច្បាស់ឬមិនសូវច្បាស់ ជាធម្មតាតទៅ GND, RS (Register Selection), RW (Read/Write)។ ពេលប្រើប្រាស់ 4 bits គេប្រើខ្សែ D4 ទៅ D7។

៣.៤.៣ របៀបតភ្ជាប់អេក្រង់ LCD ទៅ Arduino

ក្នុងរូបភាព ៣.៤៥ បង្ហាញពីការតភ្ជាប់អេក្រង់ LCD ទៅនឹង Arduino Uno។



រូបភាព ៣.៤៥ ៖ ការតភ្ជាប់អេក្រង់ LCD ទៅ Arduino Uno

គាំទ្រដោយមូលនិធិស្រាវជ្រាវ គំនិតច្នៃប្រឌិត និងនវានុវត្តន៍ (ស.គ.ន) នៃក្រសួងអប់រំ យុវជន និងកីឡា

ក្នុងរូបភាព ៣.៤៦ ជាឧទាហរណ៍ Sketch ដែលបង្ហាញពីការប្រើប្រាស់អេក្រង់ LCD ជាមួយ Arduino Uno។

```
#include<LiquidCrystal.h>

String words;

LiquidCrystal lcd( 12,10,7,6,5,4 );

void setup( )
{
   Serial.begin( 9600 );
   lcd.begin( 16,2 );
   lcd.setCursor( 0,0 );
   lcd.print( "Initialising." );
   delay( 1000 );
   lcd.clear( );
   lcd.print( "Initialising.." );
   delay( 1000 );
   lcd.clear( );
   lcd.print( "Initialising..." );
   delay( 1000 );
   lcd.clear( );
   lcd.print( "Maker : Techduino" );
   delay( 2000 );
}

void loop( )
```

```
{
   lcd.clear( );
   lcd.print( "Hello" );
   Serial.print( "Enter the line : " );
   while ( Serial.available( )==0 ) {
   }
   words = Serial.readString( );
   Serial.println( words );
   lcd.setCursor( 0,0 );
   lcd.print( words );
   delay( 5000 );
   Serial.println( " " );
}
```

រូបភាព ៣.៤៦ ៖ ឧទាហរណ៍កូដបង្ហាញព័ត៌មាននៅលើអេក្រង់ LCD

កូដខាងក្រោម LiquidCrystal library ដែលអាចឲ្យ Arduino Uno ប្រើប្រាស់អេក្រង់ LCD បាន

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,10,7,6,5,4); ប្រកាស class ដោយបញ្ជាក់ pin ដែលតភ្ជាប់នឹង "RS, En, D4, D5, D6, D7" តាមលំដាប់។

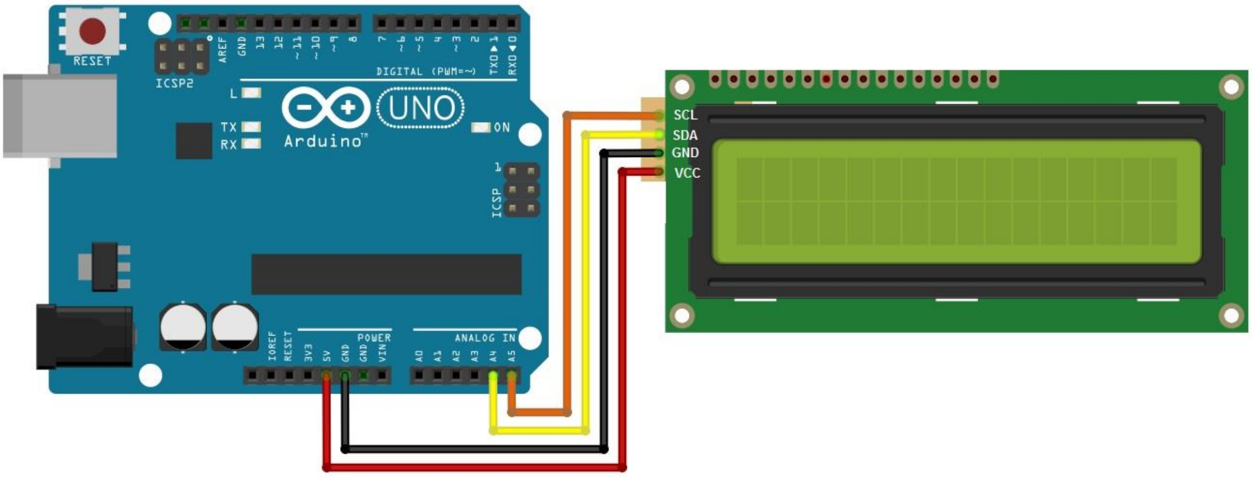
lcd.begin(16,2); ជ្រើសរើសប្រភេទ LCD 16x2។

lcd.setCursor(0,0); កំណត់កន្លែងចាប់ផ្តើមបង្ហាញអក្សរ ក្នុងករណីនេះ ជួរដេកទី១ ជួរឈរទី១ (ចំណាំ នៅក្នុងកម្មវិធី C លេខលំដាប់ចាប់ផ្តើមរាប់ពី 0 មិនមែនពី 1 ទេ)។

lcd.print("Initialising."); ព្រីន Initialising នៅលើអេក្រង់ LCD។

lcd.clear( ); លុបអក្សរទាំងអស់ ចេញពីអេក្រង់ LCD។

ក្នុងករណីប្រើប្រាស់អេក្រង់ LCD ប្រភេទ I2C 16x2 ការតភ្ជាប់ដូចបង្ហាញក្នុងរូបភាព ៣.៤៧ ដោយតភ្ជាប់ pin របស់អេក្រង់ LCD VCC ទៅ pin 5V, GND ទៅ GND, SCL ទៅ A4 និង SDA ទៅ A5 របស់ Arduino Uno តាមលំដាប់។ ក្នុងករណីនេះ ប្រើប្រាស់តែខ្សែចម្លង៤ខ្សែតែប៉ុណ្ណោះ។



រូបភាព ៣.៤៧ ៖ ការតភ្ជាប់អេក្រង់ LCD ប្រភេទ I2C ជាមួយ Arduino Uno

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd( 0x27,16,2 ); // set the LCD address to 0x27 for a 16 chars and 2 line
display
void setup( )
{
  lcd.init( ); // initialize the lcd
  lcd.init( ); // Print a message to the LCD.
  lcd.backlight( );
  lcd.setCursor( 1,0 );
  lcd.print( "hello everyone" );
  lcd.setCursor( 1,1 );
  lcd.print( "konichiwaa" );
}
void loop( )
{
```

```
}
```

រូបភាព ៣.៤៨ ៖ កូដប្រើប្រាស់អេក្រង់ LCD ប្រភេទ I2C ជាមួយ Arduino Uno

ក្នុងកូដខាងក្រោម ជាដំបូងបន្ថែម Wire និង LiquidCrystal_I2C library ដើម្បីអាចប្រើប្រាស់ Arduino Uno ជាមួយអេក្រង់ LCD ប្រភេទ I2C។

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2); ប្រកាសអញ្ញត្តិ lcd ជាប្រភេទទិន្នន័យ LiquidCrystal_I2C ដែលអាចប្រើប្រាស់ function ផ្សេងៗនៅក្នុងកូដទាក់ទងនឹងអេក្រង់ LCD នៅផ្នែកខាងក្រោមទៀត ដូចជា lcd.init( ), lcd.backlight( ), lcd.setCursor( ), lcd.print( ) ជាដើម។

lcd.init( ); សម្រាប់ចាប់ផ្តើមអេក្រង់ LCD។

lcd.backlight( ); សម្រាប់បើកភ្លើងខាងក្រោយ LCD ឲ្យភ្លឺ។

lcd.setCursor(1,0); សម្រាប់ប្តូរទីតាំងអក្សរ។

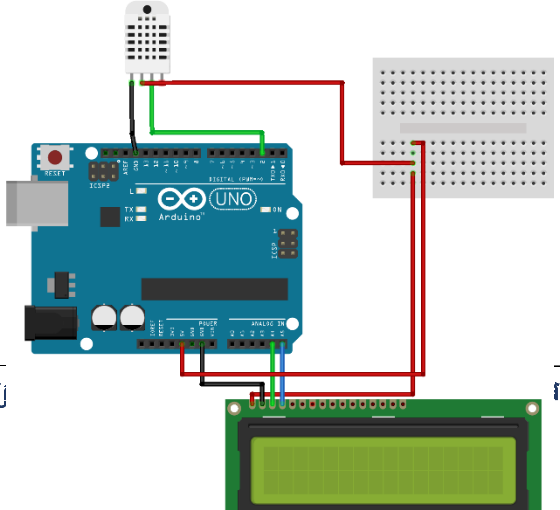
lcd.print("hello everyone"); ព្រីន hello everyone នៅលើអេក្រង់ LCD។

lcd.setCursor(1,1); សម្រាប់ប្តូរទីតាំងអក្សរ។

lcd.print("konichiwaa"); ព្រីន konichiwa នៅលើអេក្រង់ LCD។

៣.៤.៤ ការបង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសនៅលើអេក្រង់ LCD

ឧទាហរណ៍នេះ ជាការបញ្ចូលគ្នារវាងកូដសម្រាប់បង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសពីសេនស័រ DHT និងកូដដែលបង្ហាញព័ត៌មាននៅលើអេក្រង់ LCD ប្រភេទ I2C។



រូបភាព ៣.៤៩ ៖ ការតភ្ជាប់ឧបករណ៍ដើម្បីបង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសនៅលើ អេក្រង់ LCD ប្រភេទ I2C

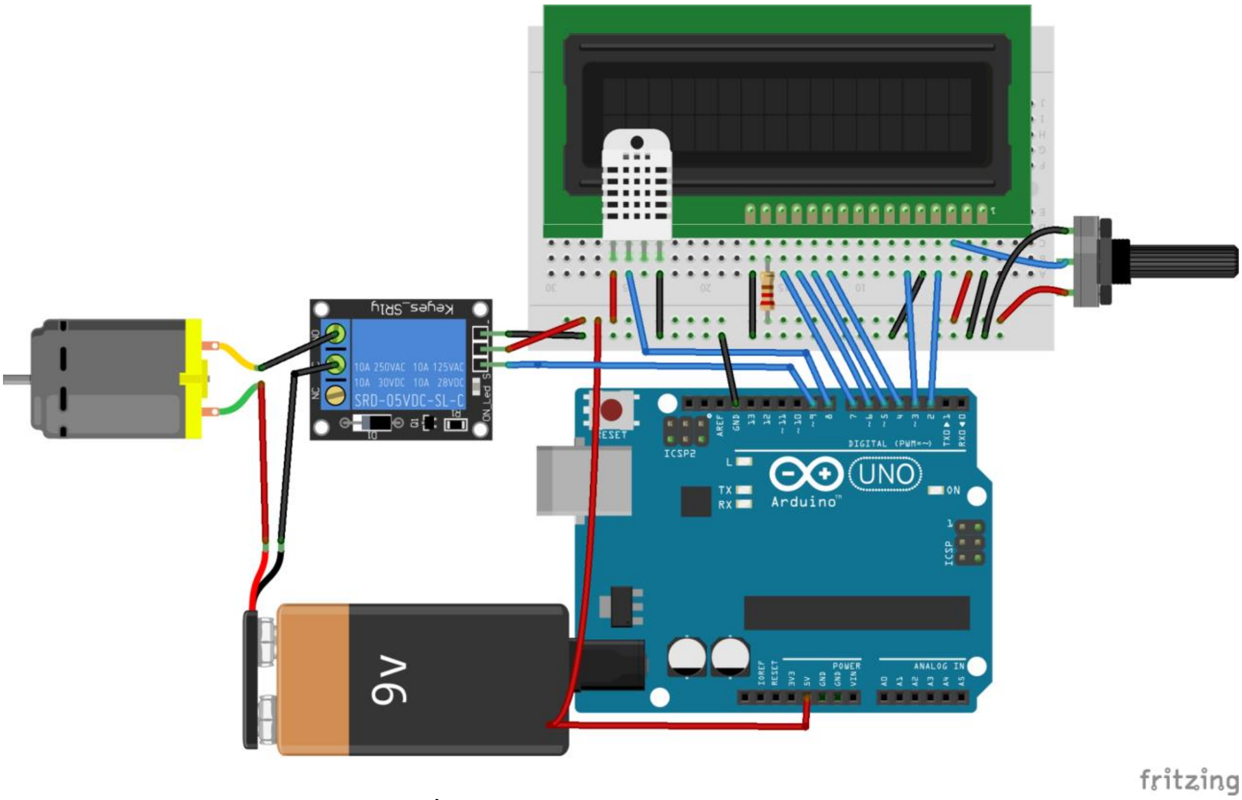
```
#include <LiquidCrystal.h>

LiquidCrystal lcd( 12, 11, 5, 4, 3, 2 );

int sensorPin = 0;


void setup( )
{
  Serial.begin( 9600 );
  lcd.begin( 16, 2 );
  pinMode( 13, OUTPUT );
  pinMode( 7, OUTPUT );
}

void loop( )
{
lcd.clear( );
int reading = analogRead( sensorPin );
float voltage = reading  *  5.0;
voltage /= 1024.0;
float temperatureC = ( voltage - 0.5 )  *  100 ;
```

```
float temperatureF = ( temperatureC  * 9.0 / 5.0 ) + 32.0;
lcd.print( temperatureF );
lcd.println( " degrees F " );
lcd.setCursor( 0,1 );
lcd.print( temperatureC );
lcd.println( " degrees C " );
delay( 1000 );
}
```

រូបភាព ៣.៥០ ៖ កូដសម្រាប់បង្ហាញសីតុណ្ហភាពនិងសំណើមបរិយាកាសនៅលើអេក្រង់ LCD

៣.៤.៥ ការបញ្ជាម៉ូទ័រអាស្រ័យលើសីតុណ្ហភាពនិងសំណើមបរិយាកាស

ប្រព័ន្ធនេះដូចបង្ហាញក្នុងរូបភាព ៣.៥១ ស្រដៀងទៅនឹងប្រព័ន្ធបញ្ជាទូភ្ញាស់សុត ដែលត្រូវរក្សា សីតុណ្ហភាពនិងសំណើមបរិយាកាសដែលសមស្របសម្រាប់ការលូតលាស់របស់ស៊ុត។ នៅក្នុងប្រព័ន្ធនេះ សីតុណ្ហភាពនិងសំណើមបរិយាកាសទទួលបានពីសេនស័រ DHT ដែលអាចប្រើប្រាស់ដើម្បីបង្ហាញព័ត៌មាននៅលើអេក្រង់ LCD ក៏ដូចជាលក្ខខណ្ឌសម្រាប់បញ្ជាឲ្យម៉ូទ័រវិល (តាមរយៈ relay module) ដែលអាចជាកង្ហារបូមខ្យល់នៅក្នុងទូភ្ញាស់ ឬ អាចជំនួសម៉ូទ័រដោយឧបករណ៍កម្តៅ (heater) សម្រាប់បង្កើនសីតុណ្ហភាពក្នុងទូភ្ញាស់នៅពេលវាចុះទាបជាងសីតុណ្ហភាពដែលបានកំណត់។



រូបភាព ៣.៥១ ៖ ការបញ្ជាម៉ូទ័រតាមលក្ខខណ្ឌសីតុណ្ហភាពនិងសំណើមបរិយាកាស

ប្រភព ៖ https://maker.pro/arduino/projects/how-to-make-a-temperature-controlled-fan-using-arduino

```
#include <LiquidCrystal.h>


LiquidCrystal lcd( 12, 11, 5, 4, 3, 2 );


int sensorPin = 0;


void setup( )
{
  Serial.begin( 9600 );
  lcd.begin( 16, 2 );
  pinMode( 13, OUTPUT );
  pinMode( 7, OUTPUT );
```

```
}

void loop( )
{
lcd.clear( );
int reading = analogRead( sensorPin );
float voltage = reading  * 5.0;
voltage /= 1024.0;
float temperatureC = ( voltage - 0.5 )  * 100 ;
float temperatureF = ( temperatureC  * 9.0 / 5.0 ) + 32.0;
lcd.print( temperatureF );
lcd.println( " degrees F " );
lcd.setCursor( 0,1 );
lcd.print( temperatureC );
lcd.println( " degrees C " );

if( temperatureC >= 30.0 )
{
  Serial.println( "Too hot" );
  digitalWrite( 13, HIGH );
  digitalWrite( 7, HIGH );
}
if( temperatureC <= 0.0 )
{
  Serial.println( "Too cold" );
  digitalWrite( 13, LOW );
  digitalWrite( 7, LOW );
}
if( temperatureC > 0 && temperatureC < 30.0 )
{
  Serial.println( "OK" );
```

```
  digitalWrite( 13, LOW );
  digitalWrite( 7, LOW );
}
delay( 1000 );


}
```

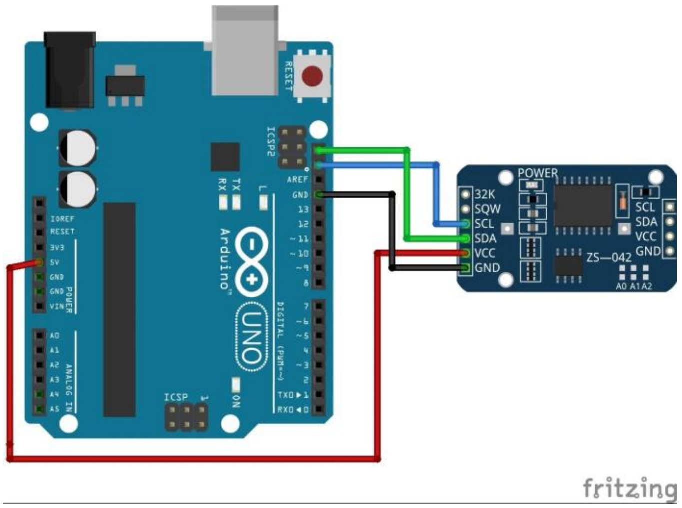
រូបភាព ៣.៥២ ៖ កូដសម្រាប់បញ្ជាម៉ូទ័រអាស្រ័យតាមលក្ខខណ្ឌសីតុណ្ហភាពនិងសំណើមបរិយាកាស

៣.៤.៦ Real-time Clock Module (DS3231)

Real-time clock module សម្រាប់ឲ្យ Arduino អាចប្រើប្រាស់កាលបរិច្ឆេទនិងម៉ោង ទោះបីយើងបិទវាក៏ដោយ ដោយសារវាមានថ្មនាឡិកា ដែលអាចឲ្យវានៅដំណើរការបាន។ នៅពេលកត់ត្រាទិន្នន័យពិសោធន៍ គេត្រូវកត់ត្រាកាលបរិច្ឆេទ និងម៉ោងដែលទិន្នន័យនេះបានកត់ត្រា ដើម្បីអាចងាយស្រួលក្នុងការយកទិន្នន័យនេះទៅវិភាគនៅពេលក្រោយ។ ការដឹងពីពេលវេលាក៏អាចប្រើប្រាស់ក្នុងការកំណត់ម៉ោង និងរយៈពេលដែលឧបករណ៍ត្រូវចាប់ផ្តើមដំណើរការ រយៈពេលដំណើរការ និងពេលបិទ។ ឧទាហរណ៍ យើងអាចកំណត់ពេលដើម្បីបើកឲ្យប្រព័ន្ធស្រោចស្រពដំណើរការ រយៈពេលដំណើរការ និងពេលដែលបិទវិញ។

រូបភាព ៣.៥៣ ៖ Real-time clock module (DS3231)

រូបភាព ៣.៥៤ បង្ហាញរបៀបតភ្ជាប់ DS3231 ទៅនឹង Arduino Uno។



រូបភាព ៣.៥៤ ៖ ការតភ្ជាប់ DS3231 ទៅនឹង Arduino Uno

កូដសម្រាប់កែសម្រួលកាលបរិច្ឆេទនិងម៉ោង ដូចបង្ហាញក្នុងរូបភាព ៣.៥៥ គេដំណើរការវាតែម្តងទេ ដោយបើក Arduino Uno ម្តង ដើម្បីកែសម្រួលកាលបរិច្ឆេទនិងម៉ោងឲ្យត្រឹមត្រូវ។

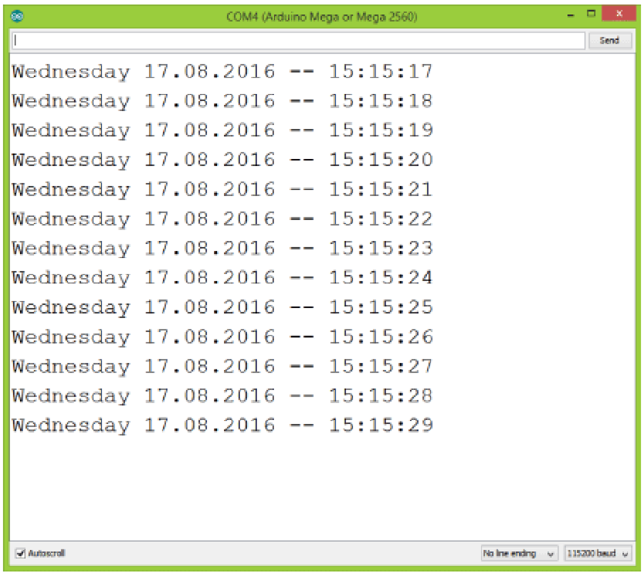
```
// Code from the Demo Example of the DS3231 Library
void setup( )
{
  // Setup Serial connection
  Serial.begin( 115200 );
  // Uncomment the next line if you are using an Arduino Leonardo
  //while ( !Serial ) {}
  // Initialize the rtc object
  rtc.begin( );
  // The following lines can be uncommented to set the date and time
  //rtc.setDOW( SUNDAY ); // Set Day-of-Week to SUNDAY
  //rtc.setTime( 12, 0, 0 ); // Set the time to 12:00:00 ( 24hr format )
  //rtc.setDate( 1, 1, 2014 ); // Set the date to January 1st, 2014
}
```

រូបភាព ៣.៥៥ ៖ កូដសម្រាប់កែសម្រួលកាលបរិច្ឆេទនិងម៉ោង

កូដក្នុងរូបភាព ៣.៥៦ បង្ហាញកាលបរិច្ឆេទនិងម៉ោងនៅលើ Serial Monitor។ នៅពេលដំណើរការវានឹងបង្ហាញលទ្ធផលដូចក្នុងរូបភាព ៣.៥៧។

```
// Code from the Demo Example of the DS3231 Library
void setup( )
{
  // Setup Serial connection
  Serial.begin( 115200 );
  // Uncomment the next line if you are using an Arduino Leonardo
  //while ( !Serial ) {}
  // Initialize the rtc object
  rtc.begin( );
  // The following lines can be uncommented to set the date and time
  //rtc.setDOW( SUNDAY ); // Set Day-of-Week to SUNDAY
  //rtc.setTime( 12, 0, 0 ); // Set the time to 12:00:00 ( 24hr format )
  //rtc.setDate( 1, 1, 2014 ); // Set the date to January 1st, 2014
}
void loop( )
{
  // Send Day-of-Week
  Serial.print( rtc.getDOWStr( ) );
  Serial.print( " " );
  // Send date
  Serial.print( rtc.getDateStr( ) );
  Serial.print( " -- " );
  // Send time
  Serial.println( rtc.getTimeStr( ) );
  // Wait one second before repeating
  delay ( 1000 );
}
```

រូបភាព ៣.៥៦ ៖ កូដសម្រាប់បង្ហាញកាលបរិច្ឆេទនិងម៉ោងនៅលើ Serial Monitor

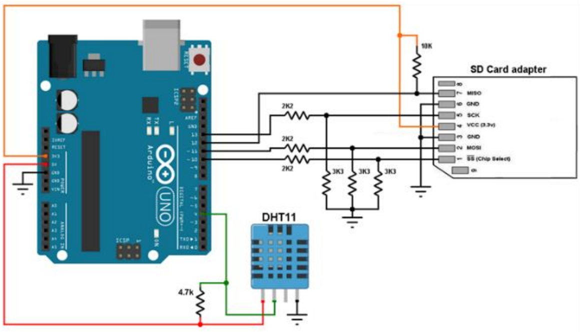


រូបភាព ៣.៥៧ ៖ Serial monitor បង្ហាញកាលបរិច្ឆេទនិងម៉ោង

៣.៤.៧ ការកាត់ត្រាទិន្នន័យនៅក្នុង SD Card

ការបង្ហាញទិន្នន័យពីសេនស័រនៅលើអេក្រង់ LCD អាចឲ្យយើងដឹងពីស្ថានភាពបច្ចុប្បន្ន ប៉ុន្តែក្នុងការពិសោធន៍ក៏ត្រូវការកត់ត្រាទិន្នន័យសម្រាប់វិភាគនៅពេលក្រោយទៀត។ គេអាចកត់ត្រាទិន្នន័យទៅក្នុង SD card ដោយប្រើប្រាស់ SD card module។ នៅក្នុងរូបភាព ៣.៥៨ ទិន្នន័យពី DHT11 ត្រូវបានកែច្នៃដោយ Arduino Uno រួចហើយកត់ត្រាទុកនៅក្នុង SD card។

សេនស័រ → Arduino → SD Card



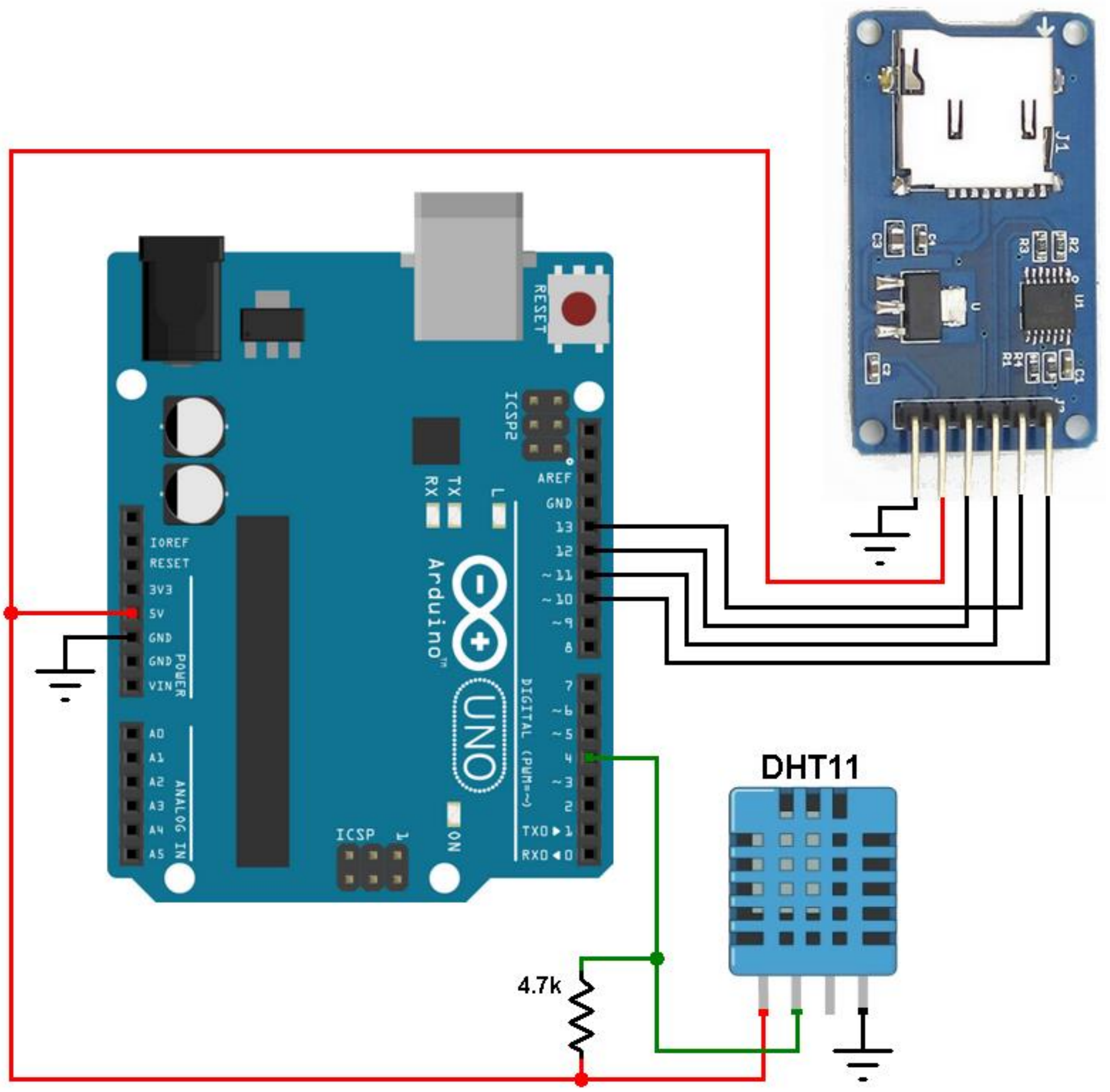
រូបភាព ៣.៥៨ ៖ ការតភ្ជាប់ឧបករណ៍ដើម្បីកាត់ត្រាទិន្នន័យពី DHT11 ទៅក្នុង SD card

ប្រភព ៖ https://simple-circuit.com/arduino-sd-card-dht11-data-logger/

ក្រៅពី SD card បច្ចុប្បន្ននេះ គេក៏អាចប្រើប្រាស់ Micro SD card ជំនួសបានដែរ ដោយសារគេនិយមប្រើប្រាស់វាសម្រាប់ផ្ទុកឯកសារ រូបថត ឬ វីដេអូ របស់ smartphone ដោយប្រើប្រាស់ SD card adapter ដូចបង្ហាញក្នុងរូបភាព ៣.៥៩។ ជាធម្មតា SD និង Micro SD ពេលទិញថ្មីមកប្រើប្រាស់ fat32 file system Partition ធំបំផុត 32 GB និងទំហំឯកសារធំបំផុត 4GB។

រូបភាព ៣.៥៩ ៖ SD card adapter សម្រាប់ប្រើជាមួយ micro SD card

រូបភាព ៣.៦០ បង្ហាញការប្រើប្រាស់ជាមួយ SD Card Module ដើម្បីកត់ត្រាទិន្នន័យសីតុណ្ហភាព និងសំណើមបរិយាកាសពីសេនស័រ DHT11។



រូបភាព ៣.៦០ ៖ ការតភ្ជាប់ SD card module ដើម្បីកត់ត្រាទិន្នន័យពី DHT11

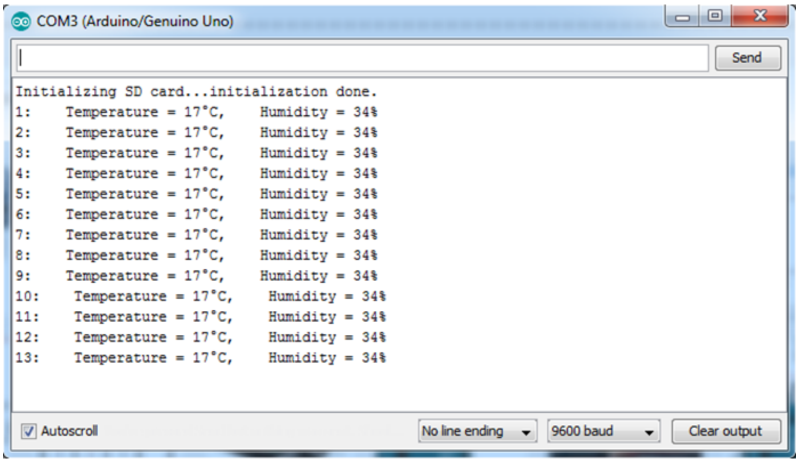
```
// Arduino data logger with SD card and DHT11 humidity and temperature sensor


#include <SPI.h>        // Include SPI library (needed for the SD card)
#include <SD.h>         // Include SD library
#include <DHT.h>         // Include DHT sensor library


File dataFile;


#define DHTPIN 4            // DHT11 data pin is connected to Arduino pin 4
#define DHTTYPE DHT11       // DHT11 sensor is used
```

```
DHT dht( DHTPIN, DHTTYPE );   // Initialize DHT library

void setup( ) {
  // Open serial communications and wait for port to open:
  Serial.begin( 9600 );
  while ( !Serial )
    ; // wait for serial port to connect. Needed for native USB port only
  Serial.print( "Initializing SD card..." );
  if ( !SD.begin( ) ) {
    Serial.println( "initialization failed !" );
    while ( 1 );
  }
  Serial.println( "initialization done." );
  delay( 2000 );
}

uint16_t line = 1;
  void loop( ) {
  delay( 1000 );
  // Read humidity
  byte RH = dht.readHumidity( );
  //Read temperature in degree Celsius
  byte Temp = dht.readTemperature( );

  dataFile = SD.open( "DHT11Log.txt", FILE_WRITE );

  // if the file opened okay, write to it:
  if ( dataFile ) {
    Serial.print( line );
    Serial.print( ":    Temperature = " );
    Serial.print( Temp );
```

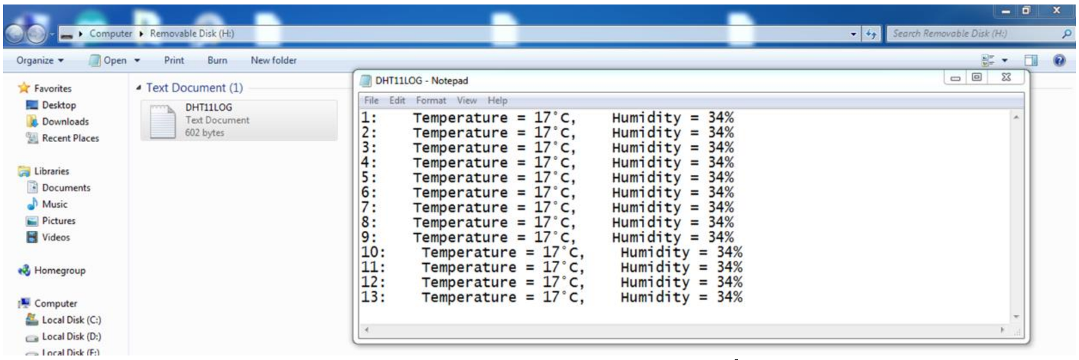
```
    Serial.print("°C,   Humidity = ");
    Serial.print(RH);
    Serial.println("%");
    // Write data to SD card file (DHT11Log.txt)
    dataFile.print(line++);
    dataFile.print(":   Temperature = ");
    dataFile.print(Temp);
    dataFile.print("°C,   Humidity = ");
    dataFile.print(RH);
    dataFile.println("%");
    dataFile.close();
  }
  // if the file didn't open, print an error:
  else
    Serial.println("error opening DHT11Log.txt");
}
```

រូបភាព ៣.៦១ ៖ កូដដើម្បីកត់ត្រាទិន្នន័យពី DHT11 ទៅក្នុង SD card module

ក្រោយពេលដំណើរការ Arduino Uno នឹងបង្ហាញលទ្ធផលក្នុង serial monitor ដូចក្នុងរូបភាព ៣.៦២ ហើយបើយើងបើកមើលក្នុង SD card នឹងឃើញឯកសារឈ្មោះ DHT11LOG.txt ដែលមានកាត់ត្រាទិន្នន័យពីសេនស័រ DHT11 នៅក្នុងនោះ ដូចបង្ហាញក្នុងរូបភាព ៣.៦៣។

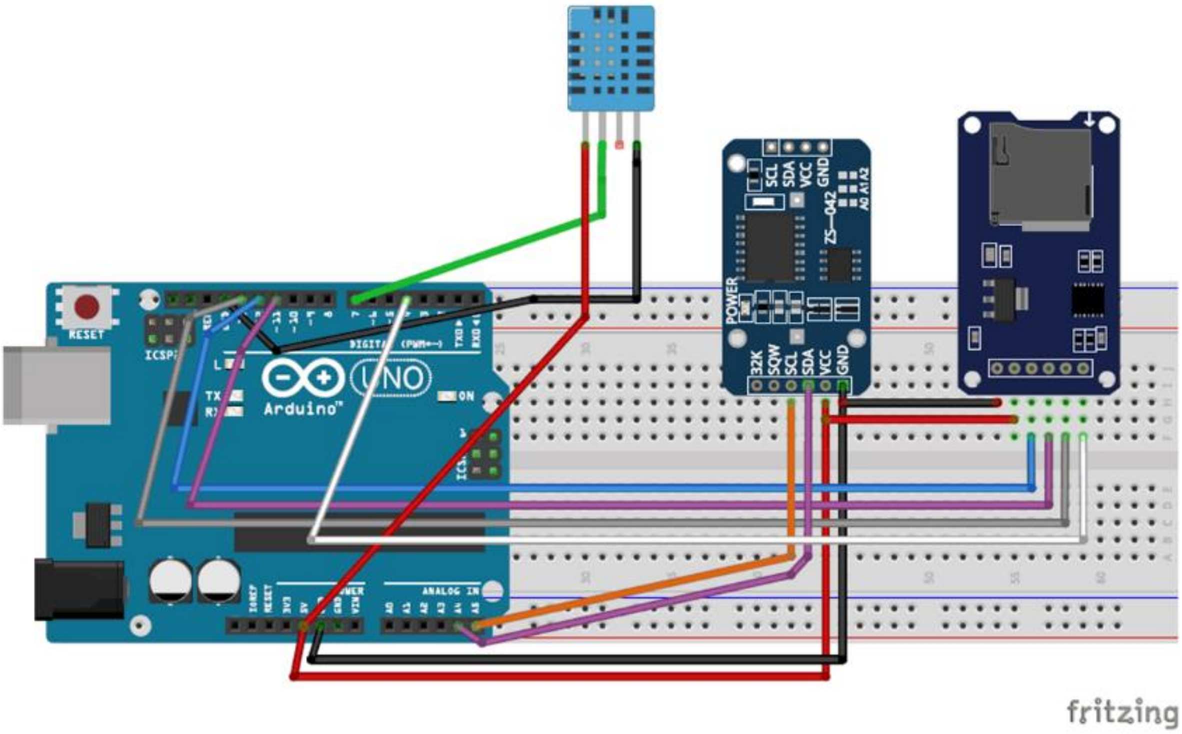


រូបភាព ៣.៦២ ៖ Serial monitor ដំណើរការកត់ត្រាទិន្នន័យ



រូបភាព ៣.៦៣ ៖ ឯកសារឈ្មោះ DHT11LOG.txt ត្រូវបានបង្កើតនៅក្នុង SD card

រូបភាព ៣.៦៤ ជាប្រព័ន្ធកត់ត្រាសីតុណ្ហភាពនិងសំណើមបរិយាកាស ទៅក្នុង SD card ដោយមានពេលវេលានិងកាលបរិច្ឆេទភ្ជាប់ជាមួយផងដែរ។ វាស្រដៀងទៅនឹងប្រព័ន្ធក្នុងរូបភាព ៣.៦០ ដែរ គ្រាន់តែមានបន្ថែម real-time clock module ដើម្បីឲ្យស្គាល់កាលបរិច្ឆេទនិងម៉ោង។

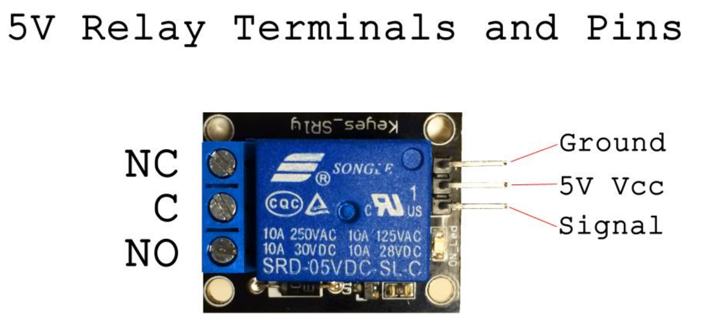


រូបភាព ៣.៦៤ ៖ ប្រព័ន្ធកត់ត្រាសីតុណ្ហភាពនិងសំណើមបរិយាកាស ទៅក្នុង SD card ដោយមានពេលវេលានិងកាលបរិច្ឆេទ

៣.៥ ការសរសេរកម្មវិធីសម្រាប់ Arduino ក្នុងបញ្ជាឧបករណ៍បញ្ចេញសកម្មភាព

៣.៥.១ Arduino Relay Module

Arduino Relay Module ត្រូវបានគេរចនាឡើងដើម្បីអាចឲ្យ Arduino ប្រើប្រាស់សម្រាប់បញ្ជា actuator បាន។ Module អាចមាន relay មួយ ឬ ច្រើន អាស្រ័យទៅតាមចំនួនឧបករណ៍ដែលចង់បញ្ជា។ ខាង input ជាធម្មតាវាមាន pin ម៉ាស (Ground) ចរន្តអគ្គីសនី 5 V (Vcc) និងសញ្ញាបញ្ជា (Signal) ដែលតភ្ជាប់ទៅនឹង digital pin របស់ ardunio ចំណែកខាង output មាន Common (C), Normally Closed (NC) និង Normally Open (NO) ដូចបង្ហាញក្នុងរូបភាព ៣.៦៥។

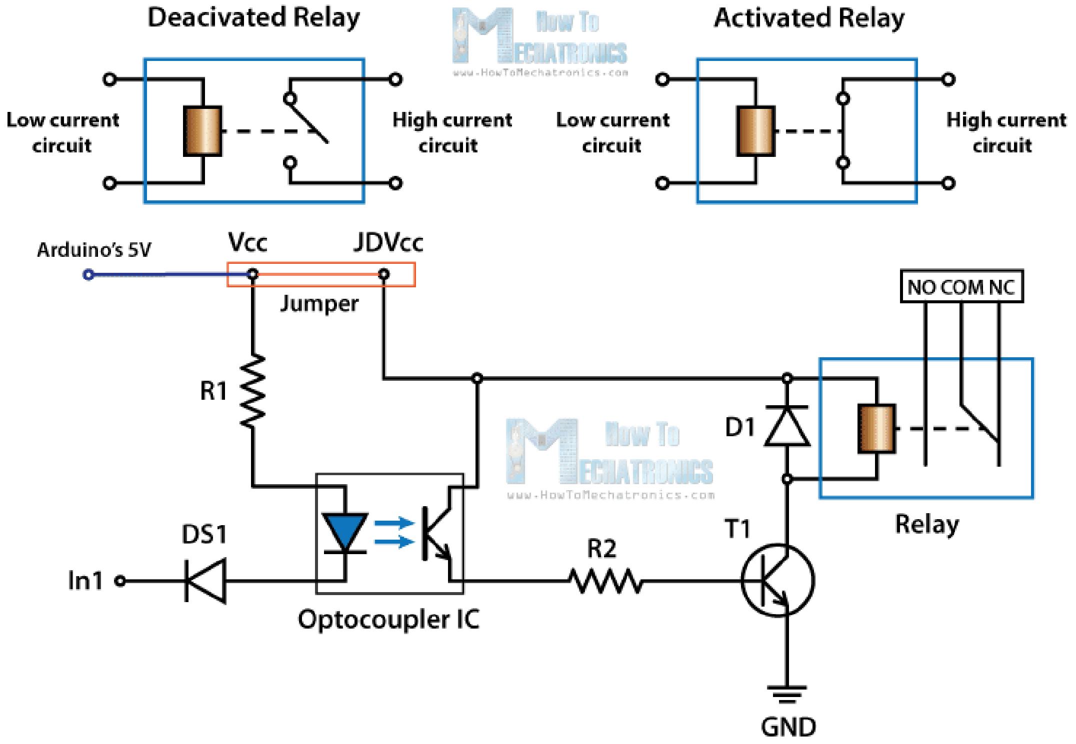


(ក)

(ខ)

(គ)

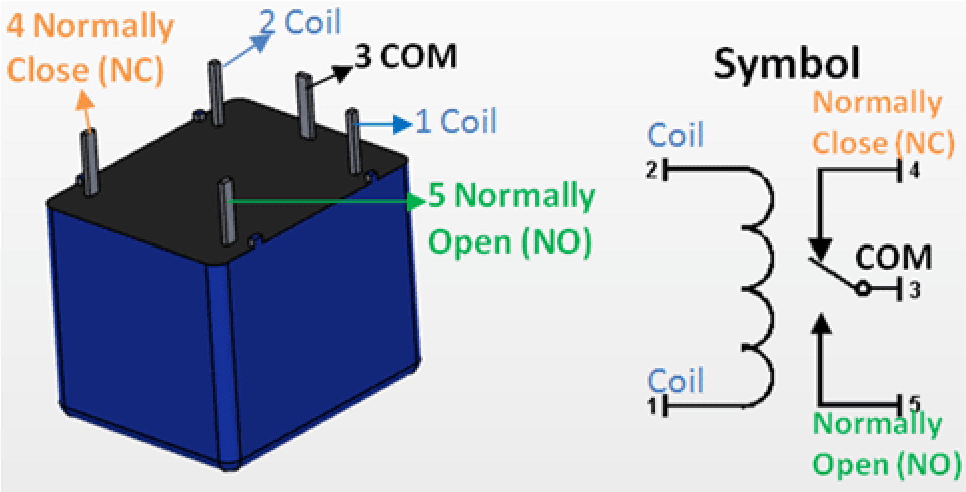
រូបភាព ៣.៦៥ ៖ Arduino Relay Module (ក) បង្ហាញ input និង output (ខ) នៅស្ថានភាព Normally Closed (គ) នៅស្ថានភាព Normally Open



រូបភាព ៣.៦៦ ៖ ធាតុសំខាន់ៗរបស់ Arduino Relay Module

ប្រភព ៖ https://www.howtomechatronics.com

នៅក្នុង module មាន Optocoupler IC ឌីយ៉ូតខាង input pin DS1, ឌីយ៉ូតភ្ជាប់ជាមួយ Relay D1 Relay ត្រង់ស៊ីស្ទ័រ T1 និងរ៉េស៊ីស្តង់ R1 និង R2 និង jumper នៅពេលតភ្ជាប់ Vcc និង JDVcc អាចផ្តល់ថាមពលអគ្គីសនីដល់ទាំង Optocoupler IC និង Relay។

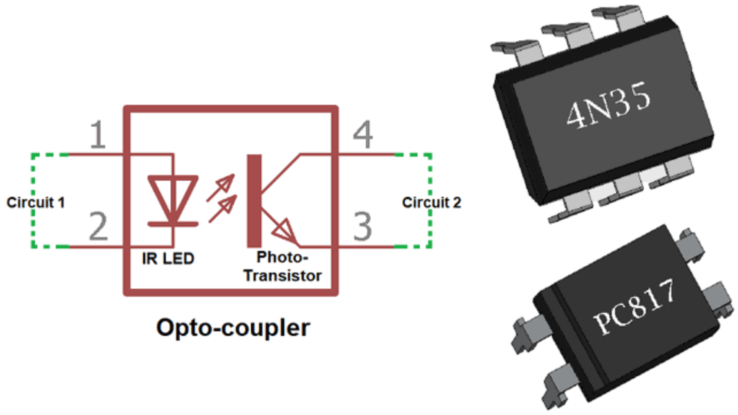


រូបភាព ៣.៦៧ ៖ Pin របស់ relay

រូបភាព ៣.៦៧ បង្ហាញ pin របស់ Relay។ Pin លេខ 1 និង 2 សម្រាប់តភ្ជាប់ទៅនឹងប្រភពចរន្ត អគ្គិសនីដែលដំណើរការអេឡិកត្រូម៉ាញ៉េទិកសម្រាប់ឲ្យ relay ដំណើរការ។ Pin លេខ 3 (COM) ជាខ្សែរួមតភ្ជាប់ទៅ actuator ចំណែក NC និង NO តភ្ជាប់ទៅ actuator តាមស្ថានភាពដែលគេចង់ឲ្យវាដំណើរការ ឬ អត់ អាស្រ័យទៅតាមស្ថានភាពរបស់អេឡិចត្រូម៉ាញ៉េទិក។ នៅពេលប្រើប្រាស់ pin លេខ 1 និង 2 ត្រូវតែជាខ្នែងជាមួយឌីយ៉ូត ដើម្បីការពារតង់ស្យុងខ្លាំងនៅក្នុងសៀគ្វីរបស់វាពេលផ្ដាច់ចរន្តពីអេឡិចត្រូម៉ាញ៉េទិក ដូចបង្ហាញនៅក្នុងរូបភាព Arduino Relay Module ខាងដើម។

៣.៥.២ Optocoupler/Opto-isolator

គេប្រើប្រាស់វាសម្រាប់បញ្ជូនសញ្ញាអគ្គីសនីរវាងសៀគ្វីពីរដាច់ពីគ្នាប្រើប្រាស់ពន្លឺ។ វាការពារតង់ស្យុងខ្ពស់កុំឲ្យមាន ឥទ្ធិពលលើប្រព័ន្ធទទួលសញ្ញាដោយសារការបញ្ជូនសញ្ញាប្រើប្រាស់ពន្លឺជំនួសឲ្យប្រើប្រាស់ខ្សែចម្លងចរន្តធម្មតា។ គេលក់វាជា IC ដូចជា 4N35 និង PC817 ដូចក្នុងរូបភាព ៣.៦៧។ វាផ្គុំឡើងដោយ LED ដែលជាប្រភពបង្កើតពន្លឺ និង Photo-transistor ជាអ្នកទទួលពន្លឺ។ នៅពេលមានចរន្តអគ្គីសនីឆ្លងកាត់ LED វាបញ្ចេញពន្លឺដែលចាំងទៅលើ base របស់ photo-transistor ធ្វើឲ្យវាចម្លងចរន្តអគ្គីសនី អាចប្រើប្រាស់ជាសញ្ញាសម្រាប់ ជា input ឧបករណ៍អេឡិកត្រូនិកផ្សេងៗ។



រូបភាព ៣.៦៨ ៖ Optocoupler/Opto-isolator

៣.៥.៣ Variable Frequency Drive (VFD)

គេប្រើ VFD ដូចបង្ហាញក្នុងរូបភាព ៣.៦៩ សម្រាប់ប្តូរល្បឿនរបស់ម៉ូទ័រចរន្តឆ្លាស់ដោយបង្កើន ឬបន្ថយប្រេកង់របស់ចរន្តអគ្គីសនី ធ្វើឲ្យល្បឿនរបស់ម៉ូទ័រកើនឡើង ឬ ថយចុះ។ ជាធម្មតា ពួកវាមានកន្លែងសម្រាប់បញ្ជា បង្ហាញព័ត៌មានពីស្ថានភាពនៅលើអេក្រង់ LCD និងកែសម្រួល (បង្កើន ឬ បន្ថយល្បឿនជុំរបស់ម៉ូទ័រ) នៅលើនោះតែម្តង ឬ គេអាចប្រើប្រាស់វាជាមួយ PLC (បញ្ជាដោយ PLC)។

រូបភាព ៣.៦៩ ៖ VFD ប្រភេទផ្សេងៗ

នៅកម្ពុជាជីវឧស្ម័នត្រូវបានប្រើប្រាស់សម្រាប់ដាំស្លនៅតាមគ្រួសារនៅជនបទ និងដំណើរការម៉ាស៊ីនភ្លើងផលិតអគ្គីសនីសម្រាប់ប្រើប្រាស់នៅតាមកសិដ្ឋានចិញ្ចឹមជ្រូក និងរោងចក្រកែច្នៃម្សៅមីជាដើម។ នៅក្នុងជីវឧស្ម័នមានជាតុមិនសុទ្ធមួយចំនួននៅលាយឡំ ដែលអាចបង្កឲ្យចង្ក្រាន និងម៉ាស៊ីនភ្លើង ងាយខូចប្រសិនបើមានបរិមាណជាតុមិនសុទ្ធទាំងនេះច្រើនហួសកម្រិតកំណត់។ រូបភាព ៣.៧០ គឺជាឧបករណ៍សម្រាប់សម្អាតជីវឧស្ម័នដោយប្រើប្រាស់គ្រាប់ដែកអុកស៊ីត សម្រាប់ដកជាតុមិនសុទ្ធនៅលាយឡំជាមួយ ជីវឧស្ម័នដូចជា ចំហាយទឹក កំទេចកំទីដី និងអ៊ីដ្រូសែនស៊ុលផួចេញ។ ឧបករណ៍នេះមានប្រើប្រាស់ VFD ដើម្បីផ្លាស់ប្តូរល្បឿនរបស់ blower ដែលបូមជីវឧស្ម័នចូលនិងឆ្លងកាត់ឧបករណ៍នេះ។ នៅពេលប្រែប្រួលល្បឿនរបស់ម៉ូទ័រធ្វើឲ្យលំហូរ និងសំពាធរបស់ជីវឧស្ម័នក៏ប្រែប្រួលតាមនេះដែរ។

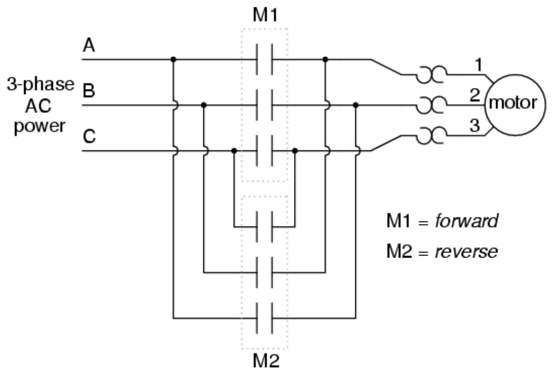
រូបភាព ៣.៧០ ៖ ប្រព័ន្ធច្រោះជីវឧស្ម័ន (Desulfurization System)

រូបភាព ៣.៧១ ៖ ការធ្វើតេស្តម៉ាស៊ីនកិនស្រូវខ្នាតតូច

រូបភាព ៣.៧១ គឺជាម៉ាស៊ីនកិន និងសម្រិតអង្ករ ដែលមានអូសទាញដោយម៉ូទ័របីហ្វា។ តាមធម្មតា ល្បឿនរបស់ម៉ូទ័រមិនចាំបាច់ផ្លាស់ប្តូរទេ ប៉ុន្តែនៅក្នុងករណីប្រើប្រាស់ម៉ាស៊ីននេះ ដើម្បីធ្វើតេស្តកិនស្រូវប្រភេទផ្សេងៗគ្នា គេប្រើប្រាស់ VFD ដើម្បីបង្កើន ឬ បន្ថយល្បឿនជុំវិលរបស់ម៉ូទ័រ ដើម្បីរកល្បឿនកិនសមស្របសម្រាប់ប្រភេទស្រូវផ្សេងៗគ្នា។

៣.៥.៤ បញ្ជាម៉ូទ័របីហ្វាប្រើប្រាស់ Contactor

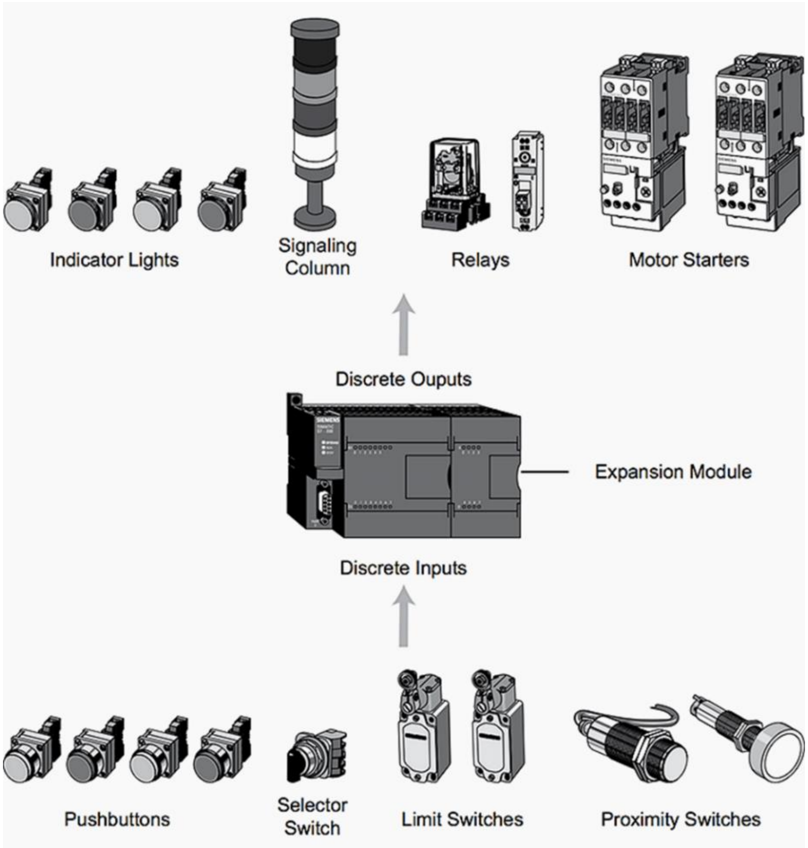
ការបញ្ជាម៉ូទ័រនៅក្នុងវិស័យកសិកម្ម និងឧស្សាហកម្ម គេច្រើនប្រើប្រាស់ contactor ដើម្បីបើកបិទម៉ូទ័រ ឬ ប្រើប្រាស់ contactor ច្រើនបញ្ចូលគ្នា ដើម្បីប្តូរទិសដៅវិលរបស់ម៉ូទ័រជាដើម ឧទាហរណ៍ ក្នុងម៉ាស៊ីនក្រឡឹង ម៉ូទ័រអាចវិលតាមទិសដៅស្រប និងបញ្ច្រាស អាស្រ័យលើតម្រូវការប្រើប្រាស់។ ដំណើរការរបស់វាស្រដៀងគ្នាទៅនឹង relay ដែរ គ្រាន់តែ contactor មានភាពសាំញ៉ាំជាងបន្តិច។ នៅក្នុងរូបភាព ៣.៧២ នៅពេល contactor M1 បិទ ម៉ូទ័រវិលតាមទិសដៅស្រប ដោយវាភ្ជាប់ខ្សែផ្គត់ផ្គង់ចរន្ត A, B, C ទៅតំណខ្សែម៉ូទ័រ 1, 2, 3 តាមលំដាប់។ ផ្ទុយមកវិញបើ contactor M2 បិទ ម៉ូទ័រនឹងវិលបញ្ច្រាសវិញ ដោយសារវាប្តូរពីខ្សែផ្គត់ផ្គង់ចរន្ត A ទៅតំណ 2 របស់ម៉ូទ័រ ហើយខ្សែផ្គត់ផ្គង់ចរន្ត B ប្តូរទៅតភ្ជាប់តំណ 1 របស់ម៉ូទ័រ ចំណែកខ្សែផ្គត់ផ្គង់ចរន្ត C នៅតភ្ជាប់ទៅតំណ 3 ដដែល។ ដើម្បីសុវត្ថិភាព ម៉ូទ័រតែងមានឧបករណ៍ការពារនៅពេលម៉ូទ័រលើសបន្ទុក (គាំង) ដើម្បីការពារកុំឲ្យឆេះខូចខាតម៉ូទ័រ ឬ អាចបង្កជាអគ្គីភ័យផងដែរ។ ប៊ូតុងសម្រាប់បញ្ជាឲ្យម៉ូទ័រវិលស្រប វិលបញ្ច្រាស ឬ ឈប់វិល ផ្គត់ផ្គង់ថាមពលដោយចរន្តដែលមានតង់ស្យុង ២៤ វ៉ុល។



រូបភាព ៣.៧២ ៖ ការប្រើប្រាស់ contactor ដើម្បីផ្លាស់ប្តូរទិសដៅវិលរបស់ម៉ូទ័របីហ្វា

ក្រៅពីប៊ូតុង នៅមានឧបករណ៍មួយចំនួនទៀតដូចក្នុងរូបភាព ៣.៧៣ សម្រាប់ផ្តល់ជាតុចូល និងជាតុចេញពី PLC ឬ ដុំបន្ថែមមុខងាររបស់វា (Expansion Module) ដែលប្រើប្រាស់ក្នុងប្រព័ន្ធបញ្ជាដូចជាភ្លើងសញ្ញាបង្ហាញ កុងតាក់ជ្រើសរើស។ល។ ឧបករណ៍ផ្តល់ជាតុចូល មានប៊ូតុងចុច (Pushbuttons) ប៊ូតុងជ្រើសរើស (Selector Buttons) កុងតាក់កំណត់កម្រិត (Limit Switches) និងកុងតាក់ដឹងរបស់មកជិតដល់ (Proximity Switches)។ ចំណែកឧបករណ៍ជាតុចេញមាន អំពូលភ្លើងសញ្ញា អំពូលបង្ហាញកម្រិត Relay និង ឧបករណ៍ចាប់ផ្តើមដំណើរការម៉ូទ័រ (Motor Starter)។ ឧបករណ៍ចាប់ផ្តើមដំណើរការម៉ូទ័រជួយ

ឲ្យម៉ូទ័រចាប់ផ្តើមដំណើរការដោយរលូន មិនទាញចរន្តអគ្គីសនីខ្លាំងពេកពីបណ្តាញ ធ្វើឲ្យរំខានដល់ឧបករណ៍អគ្គីសនីផ្សេងទៀតដែលកំពុងតភ្ជាប់ទៅនឹងបណ្តាញចរន្តអគ្គីសនីជាមួយគ្នា។



រូបភាព ៣.៧៣ ៖ ឧបករណ៍ប្រើប្រាស់ជាមួយ PLC

៣.៦ ការបញ្ជា Arduino ពីចម្ងាយតាមរយៈប្រព័ន្ធអ៊ីនធ័រណេត

៣.៦.១ ការណែនាំពីប្រព័ន្ធអ៊ីនធ័រណេត

អ៊ីនធ័រណេតជាប្រព័ន្ធតភ្ជាប់កុំព្យូទ័រនៅលើពិភពលោក ដែលអាចឲ្យកុំព្យូទ័រទំនាក់ទំនងគ្នាដើម្បីផ្ញើនឹងទទួលព័ត៌មានពីគ្នាទៅវិញទៅមក។ គេអាចប្រើប្រាស់វាដើម្បីបើកមើលវេបសាយ ផ្ញើសារអេឡិកត្រូនិក បណ្តាញទំនាក់ទំនងសង្គម ផ្ញើសំលេង និងមើលវីដេអូជាដើម។ វាជាបណ្តាញប្រភេទ TCP/IP ដែលអាចតភ្ជាប់ដោយប្រើប្រាស់ខ្សែ ឬ WiFi។ ខាងក្រោមនេះ ជាវាក្យសព្ទមួយចំនួនដែលជួយឲ្យយើងអាចយល់ពីរបៀបប្រើប្រាស់ microcontroller ជាមួយ Internet។

Server ៖ ជាកុំព្យូទ័រ ឬ microcontroller ផ្តល់សេវាកម្មអ៊ីនធ័រណេតដល់គេ ឧទាហរណ៍ web server, mail server ជាដើម។

Client ៖ ជាកុំព្យូទ័រ ឬ microcontroller សម្រាប់ប្រើប្រាស់សេវាកម្មរបស់ server ឧទាហរណ៍ web client ឬ web browser, e-mail client ជាដើម។

SSID (Service Set Identifier) ៖ ជាឈ្មោះរបស់ WiFi។

IP address ៖ គឺជាលេខដែលមានទម្រង់ 192.168.0.123 ឬ 127.0.0.1 សម្រាប់សម្គាល់កុំព្យូទ័រនីមួយៗដែលតភ្ជាប់ទៅនឹង local area network ឬ អ៊ីនធឺណេត ប្រៀបដូចទៅនឹងលេខទូរស័ព្ទដែរ។ វាអាចជា public IP address ដែលគេអាចតភ្ជាប់ទៅនឹងអ៊ីនធឺណេតបាន។ កុំព្យូទ័រនៅក្នុងបណ្តាញតែមួយ មិនអាចមានលេខ IP address ដូចគ្នាទេ មិនដូច្នេះទេ នឹងមានការរំខាន ឬ ច្រឡំនៅផ្ញើ និងទទួលព័ត៌មាន។

URL (Uniform Resource Locator) ៖ ឧទាហរណ៍ http://www.rua.edu.kh ជា URL នៃ web server របស់សាកលវិទ្យាល័យភូមិន្ទកសិកម្ម។ គេអាចបើក web server ដោយប្រើប្រាស់ public IP address ឬ ប្រើប្រាស់ URL ដែលមានភាពខាងស្រួលចងចាំជាងប្រើប្រាស់ IP address។

Port ៖ ជាលេខដែលគេបន្ថែមនៅខាងក្រោយ IP address ឬ URL របស់ server ឧទាហរណ៍ http://www.rua.edu.kh:80 ដែល 80 ជា port របស់ web server។ ជាធម្មតា គេមិនចាំបាច់ដាក់វាក៏បាន។

HTTP (HyperText Transfer Protocol) ៖ ជា protocol សម្រាប់ទំនាក់ទំនងគ្នារបស់ web server និង web browser។ គេដាក់វានៅផ្នែកខាងមុខរបស់ URL ឧទាហរណ៍ http, https, ftp។ល។

HTML (HyperText Markup Language) ៖ ជាភាសាសម្រាប់បង្ហាញ web page។

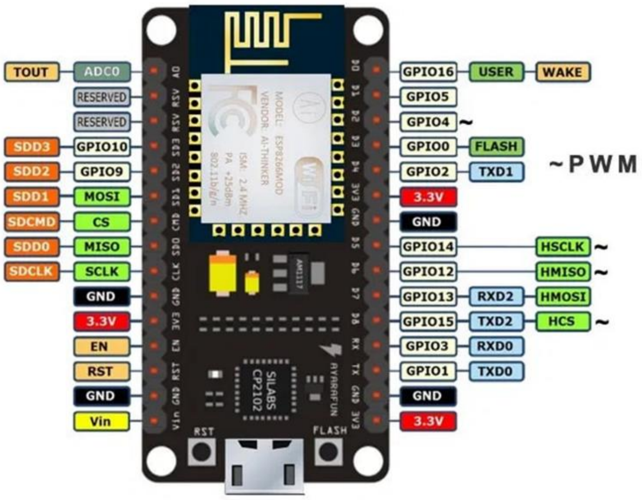
DHCP (Dynamic Host Configuration Protocol) ៖ គឺជា protocol សម្រាប់ផ្តល់ IP address ដល់កុំព្យូទ័រដែលតភ្ជាប់ទៅនឹង network ដោយស្វ័យប្រវត្តិ។ នៅពេលយើងតភ្ជាប់ទៅ router ឬ WiFi router វានឹងផ្តល់ IP address ដល់កុំព្យូទ័រយើងដោយស្វ័យប្រវត្តិ។

Router ៖ ជាឧបករណ៍ដែលអាចឲ្យកុំព្យូទ័រនៅក្នុង network ពីរ តភ្ជាប់ទៅវាអាចទំនាក់ទំនងគ្នា បាន។ ឧទាហរណ៍ WiFi router អាចឲ្យកុំព្យូទ័រតភ្ជាប់ទៅវា (local area network) តភ្ជាប់ទៅ network ខាងក្រៅ (អ៊ីនធឺណេត)។

៣.៦.២ NodeMCU

ជាប្រភេទ microcontroller ដែលអាចតភ្ជាប់ទៅនឹងប្រព័ន្ធអ៊ីនធឺណេតតាមរយៈ WiFi ដែលអាចឲ្យវាផ្ញើនិងទទួលព័ត៌មាន ឬ បញ្ជាតាមប្រព័ន្ធអ៊ីនធឺណេត។ អាស្រ័យលើកម្មវិធី (sketch) របស់វា វាអាចដើរតួនាទីជា web client/server។ ស្រដៀងទៅនឹង Arduino ដែរ គេអាចប្រើប្រាស់

Arduino IDE ដើម្បី សរសេរកម្មវិធីសម្រាប់វា ប៉ុន្តែគេត្រូវបន្ថែម board របស់ NodeMCU ទៅក្នុង Arduino IDE។

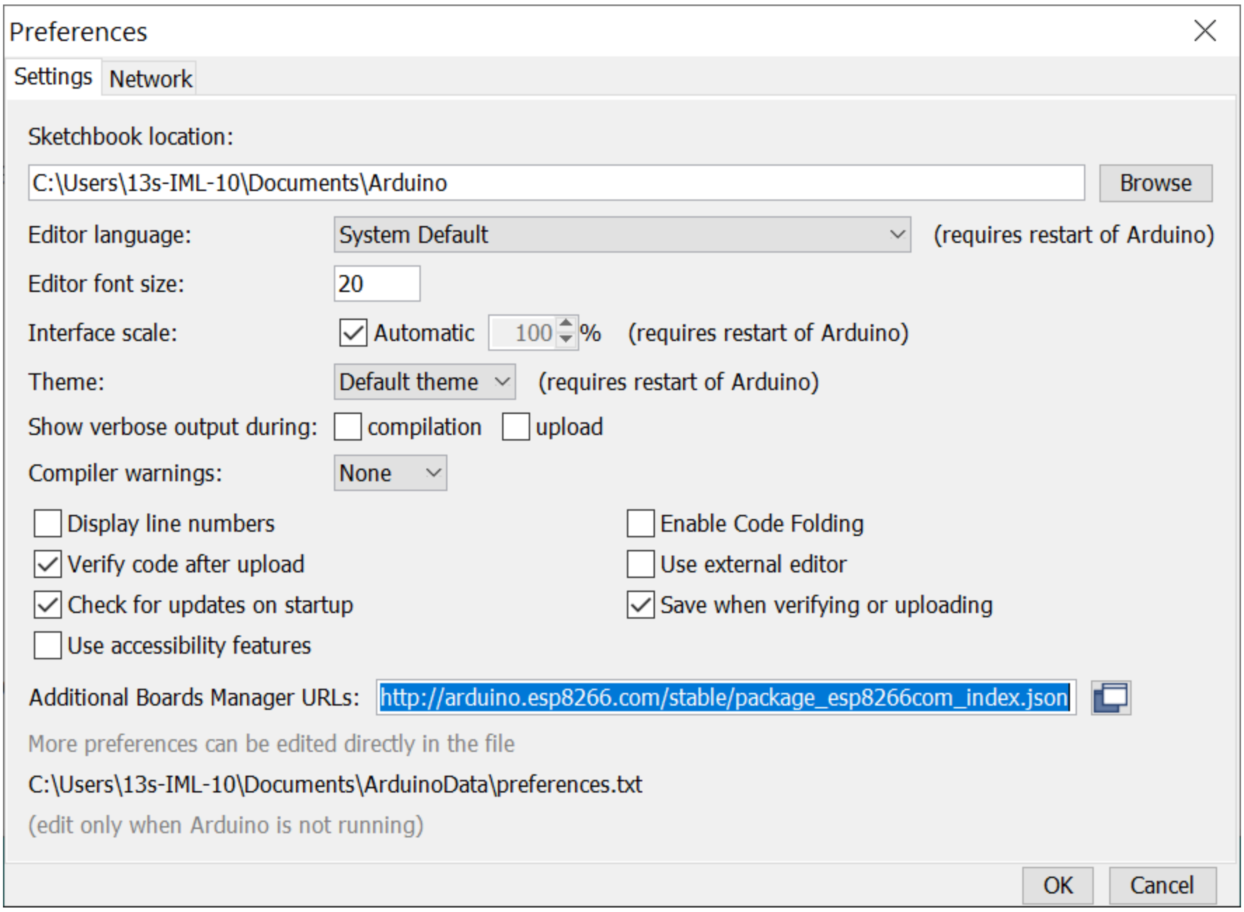


រូបភាព ៣.៧៤ ៖ NodeMCU ដែលបង្ហាញពី pin របស់វា

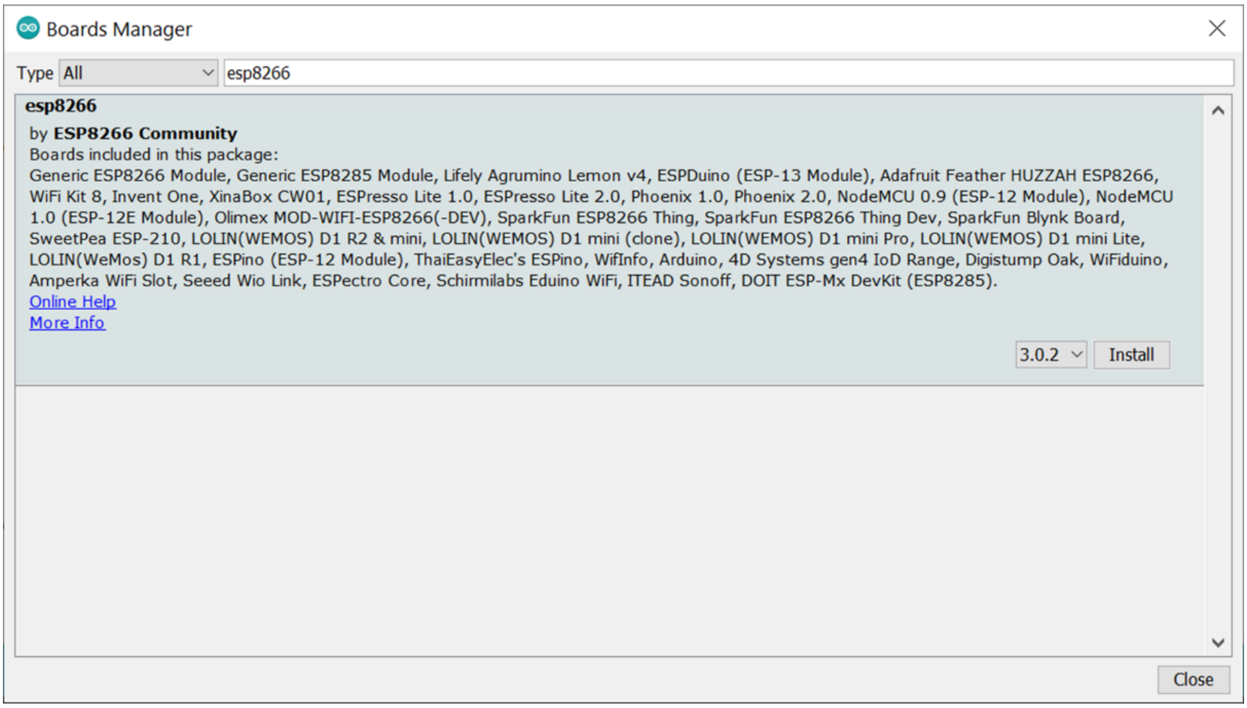
៣.៦.៣ បន្ថែម NodeMCU Board

១. បើក Arduino IDE។ បន្ទាប់មកចុច Files -> Preferences។ វាយ http://arduino.esp8266.com/stable/package_esp8266com_index.json នៅកន្លែង Additional Board Manager រួចចុច OK សូមមើលរូបភាព ៣.៧៥។

២. ចុច Tools -> Boards -> Board Manager ស្វែងរក ESP8266 រួចចុច Install ដើម្បីបន្ថែម NodeMCU Board សូមមើលរូបភាព ៣.៧៦។



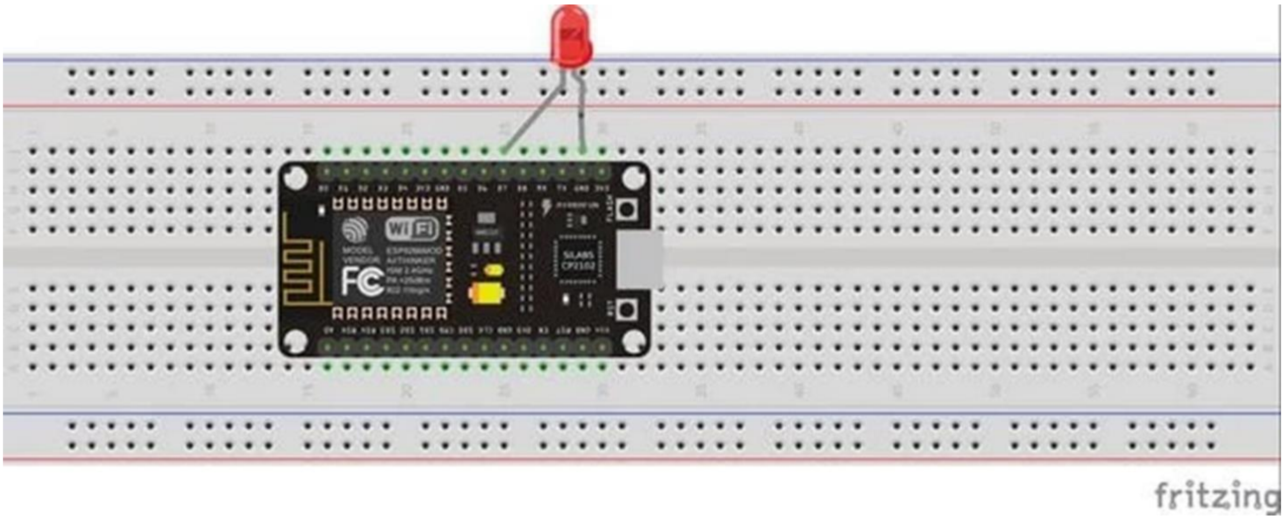
រូបភាព ៣.៧៥ ៖ Preference dialog box របស់ Ardunio IDE ពេលបន្ថែម nodeICU



រូបភាព ៣.៧៦៖ Boards Manager dialog box នៅពេល install ESP8266

៣.៦.៨ ឧទាហរណ៍ការប្រើប្រាស់ NodeMCU

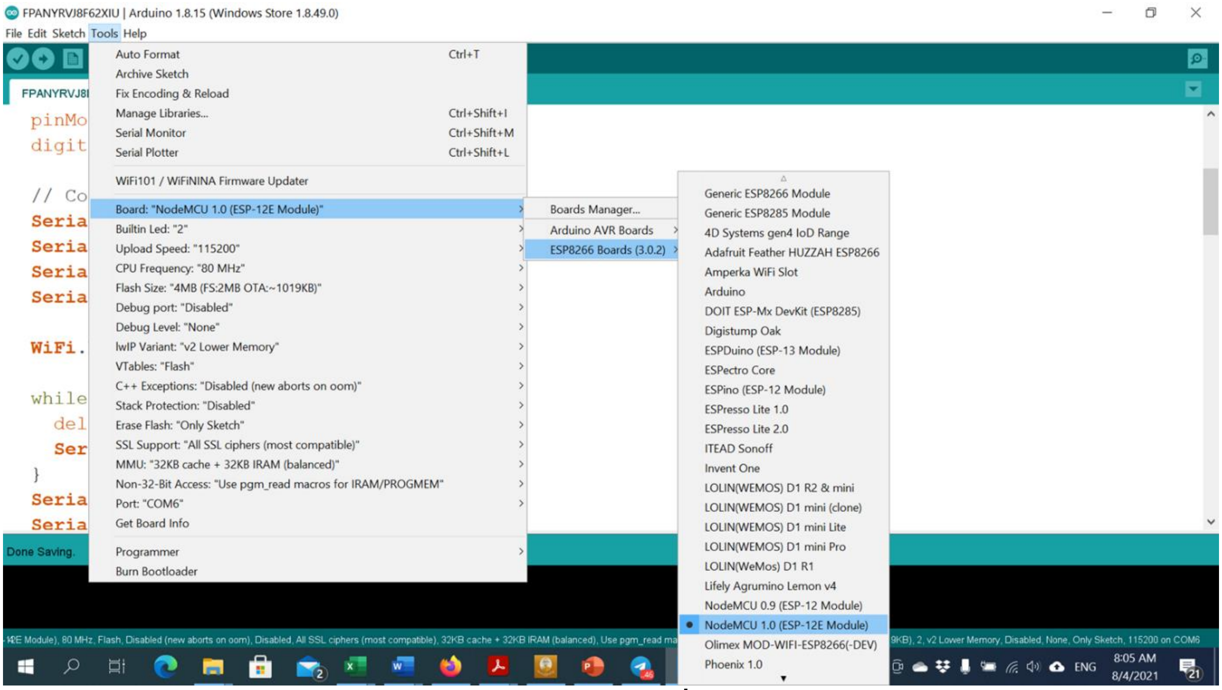
ដំបូងត Anode របស់ LED ទៅ D7 របស់ NodeMCU និង Cathode របស់ LED ទៅ GND របស់ NodeMCU សូមមើលរូបភាព ៣.៧៧ ។



រូបភាព ៣.៧៧ ៖ ការតភ្ជាប់ LED ទៅនឹង NodeMCU

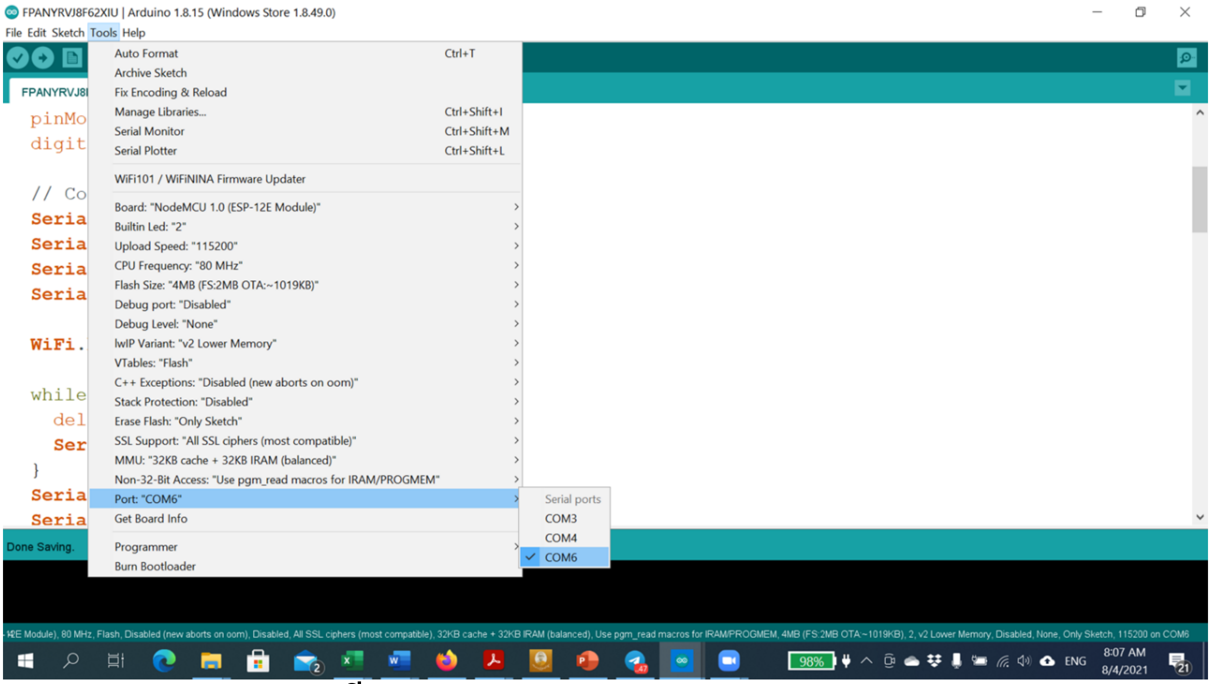
ប្រភព ៖ https://www.instructables.com/Internet-Controlled-LED-Using-NodeMCU/

បន្ទាប់មកទៀត នៅក្នុង Arduino IDE ជ្រើសរើសប្រភេទ NodeMCU ដូចបង្ហាញក្នុងរូបភាព ៣.៧៨។



រូបភាព ៣.៧៨ ៖ ការជ្រើសរើសប្រភេទ NodeMCU

រួចហើយជ្រើសរើស port ដូចបង្ហាញក្នុងរូបភាព ៣.៧៩។



រូបភាព ៣.៧៩ ៖ ការជ្រើសរើសប្រភេទ NodeMCU

បន្ថែមកូដក្នុងរូបភាព ៣.៨០ ទៅក្នុង Arduino IDE។

```
#include <ESP8266WiFi.h>

const char* ssid = “WIFI”;
const char* password = “12345”;

int ledPin = 13; // GPIO13---D7 of NodeMCU
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
```

```
  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.print("Use this URL to connect: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String request = client.readStringUntil('\r');
  Serial.println(request);
  client.flush();
```

```
  // Match the request

  int value = LOW;
  if (request.indexOf("/LED=ON") != -1)  {
    digitalWrite(ledPin, HIGH);
    value = HIGH;
  }
  if (request.indexOf("/LED=OFF") != -1)  {
    digitalWrite(ledPin, LOW);
    value = LOW;
  }

// Set ledPin according to the request
//digitalWrite(ledPin, value);

  // Return the response
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println(""); //  do not forget this one
  client.println("<!DOCTYPE HTML>");
  client.println("<html>");

  client.print("Led is now: ");

  if(value == HIGH) {
    client.print("On");
  } else {
    client.print("Off");
  }
  client.println("<br><br>");
```

```
client.println("<a href=\"/LED=ON\"\"><button>On </button></a>");
client.println("<a href=\"/LED=OFF\"\"><button>Off </button></a><br />");
client.println("</html>");

delay(1);
Serial.println("Client disonnected");
Serial.println("");

}
```

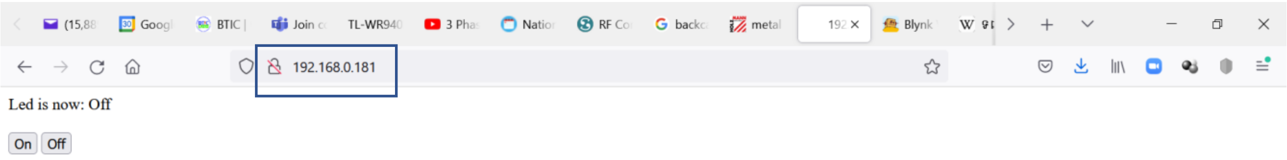
រូបភាព ៣.៨០ ៖ កូដសម្រាប់បើកបិទអំពូល LED តាមរយៈ web browser

កែកូដ (sketch) ដូចខាងក្រោម មុននឹង upload ទៅក្នុង NodeMCU ដើម្បីឲ្យដំណើរការ។ ប្តូរកូដពីរជួរខាងក្រោម

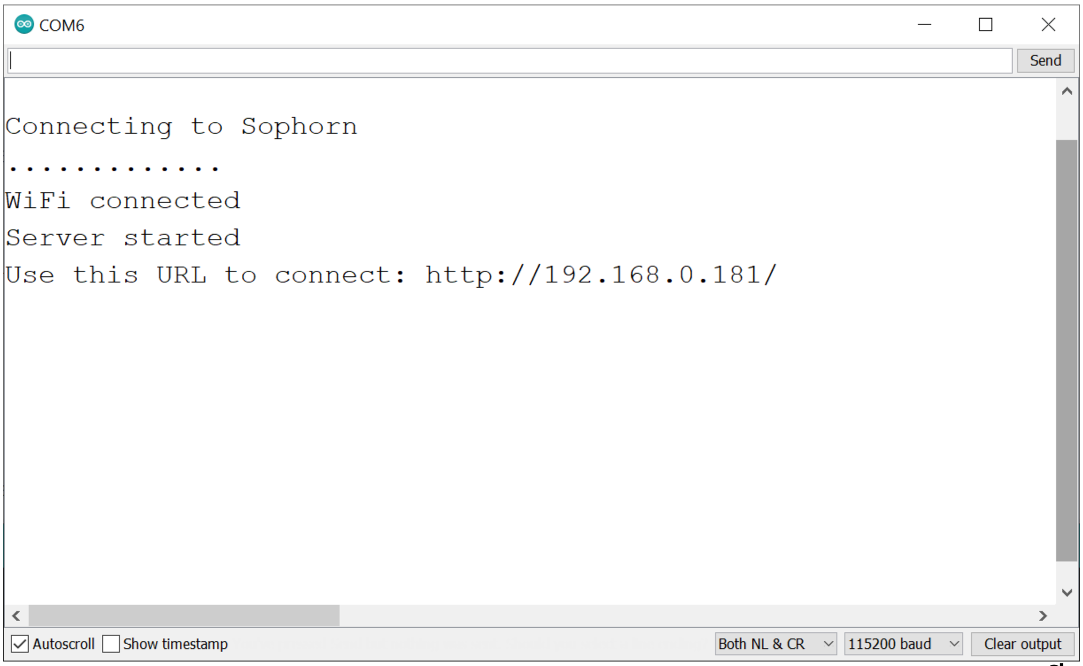
```
const char* ssid = "WIFI";
const char* password = "12345";
```

ដោយ "WIFI" ជាឈ្មោះរបស់ WiFi និង "12345" ជា password របស់ WiFi។

ក្នុងករណីនេះ NodeMCU ដើរតួនាទីជា web server ដែលគេអាចចូលមើលវាបានដោយវាយ IP address នៅ address bar របស់ web browser ដូចជា Firefox, Chrome, Safari, Microsoft Edge។ វានឹងបង្ហាញ page ដូចក្នុងរូបភាព ៣.៨១។ គេអាចដឹង IP address របស់វា ដោយមើលនៅក្នុង Serial Monitor ដូចបង្ហាញក្នុងរូបភាព ៣.៨២។ ចុច On LED នឹងភ្លឺ បិទវាវិញដោយចុច Off។ ក្នុងករណីនេះ គេអាចបើក web server តែពីកុំព្យូទ័រ ឬ smartphone ដែលតភ្ជាប់ទៅនឹង local area network ដូចគ្នាទៅនឹង NodeMCU តែប៉ុណ្ណោះ។ ប្រសិនបើប្រើប្រាស់ public IP address គេអាចបញ្ជាបើកបិទភ្លើងរបស់ LED ពីទីណាក៏បានឲ្យតែអាចតភ្ជាប់ទៅនឹងអ៊ីនធ័រណេតបាន។



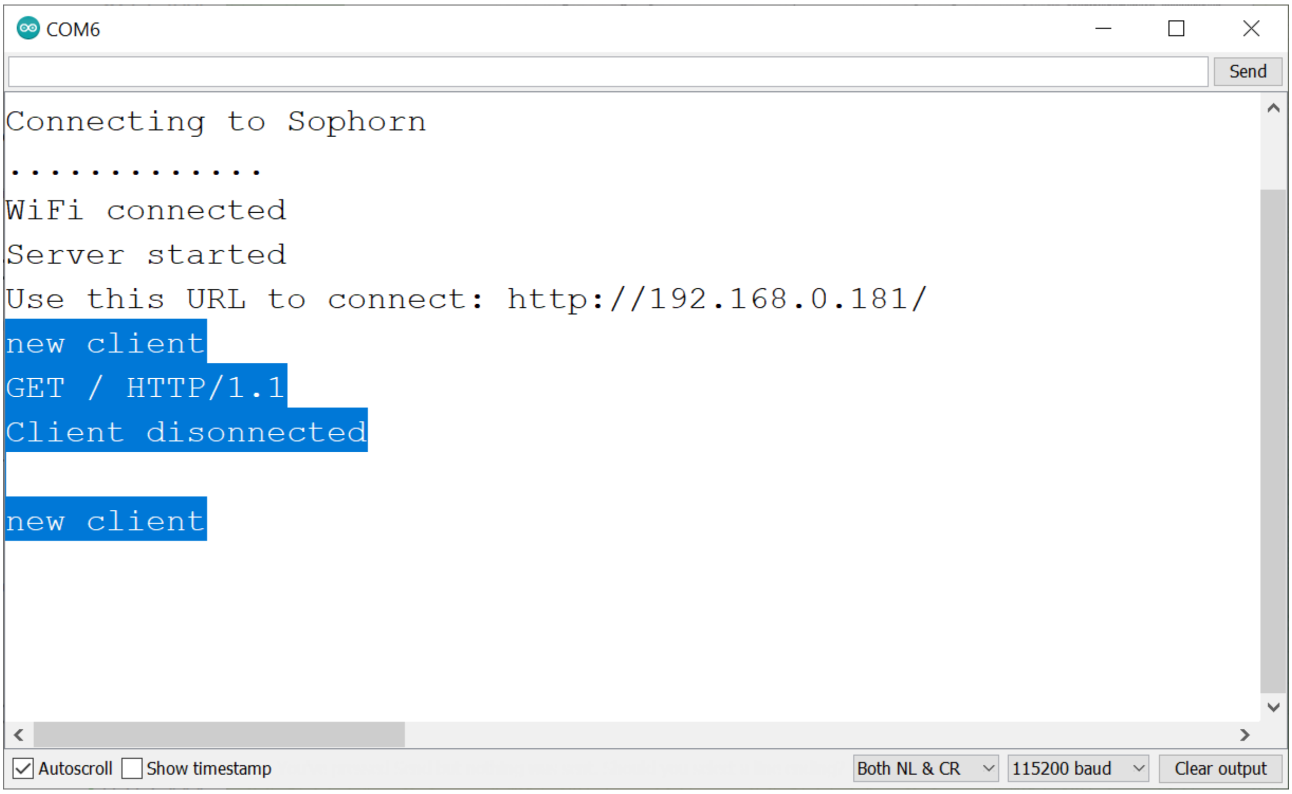
រូបភាព ៣.៨១ ៖ Web browser បង្ហាញ page ពី NodeMCU ដែលអាចបញ្ជាបើកបិទ LED



រូបភាព ៣.៨២ ៖ Serial monitor បង្ហាញ NodeMCU ដើរតួនាទីជា server សម្រាប់បញ្ជា LED

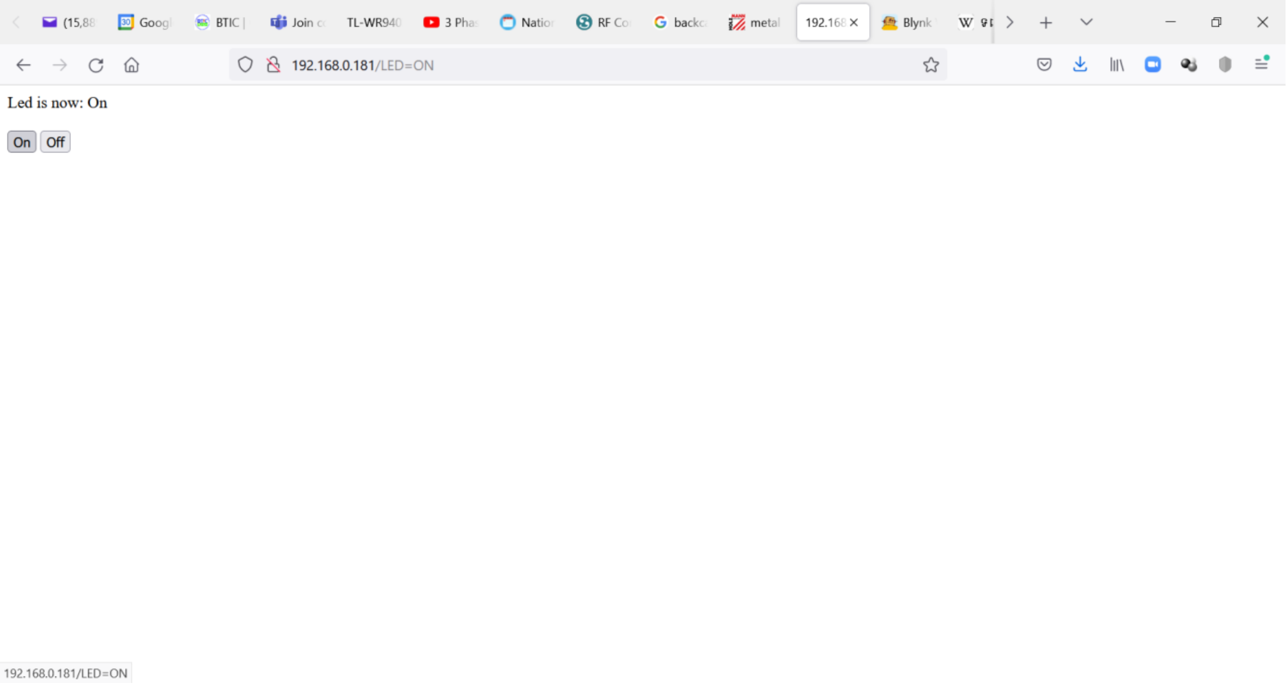
នៅក្នុងរូបភាព ៣.៨២ បង្ហាញ Serial Monitor របស់ Arduino IDE នៅពេលដំណើរកូដរបស់ NodeMCU។ ដំបូងវាបង្ហាញពីដំណើរការតភ្ជាប់ទៅ WiFi ឈ្មោះ Sophorn។ ក្រោយ

ពីតភ្ជាប់រួចហើយ វាចាប់ផ្ដើមដំណើរការ web server ហើយបង្ហាញ IP address (192.168.0.181) ដែលគេអាចប្រើប្រាស់ដើម្បីបើកចូលបញ្ជាវាបានពី web browser។ នៅពេលគេចូលតាម web browser នៅក្នុង Serial Monitor វានឹងបង្ហាញព័ត៌មាន new client បង្ហាញថាមានគេតភ្ជាប់ទៅ server ដូចបង្ហាញដូចក្នុងរូបភាព ៣.៨៣។

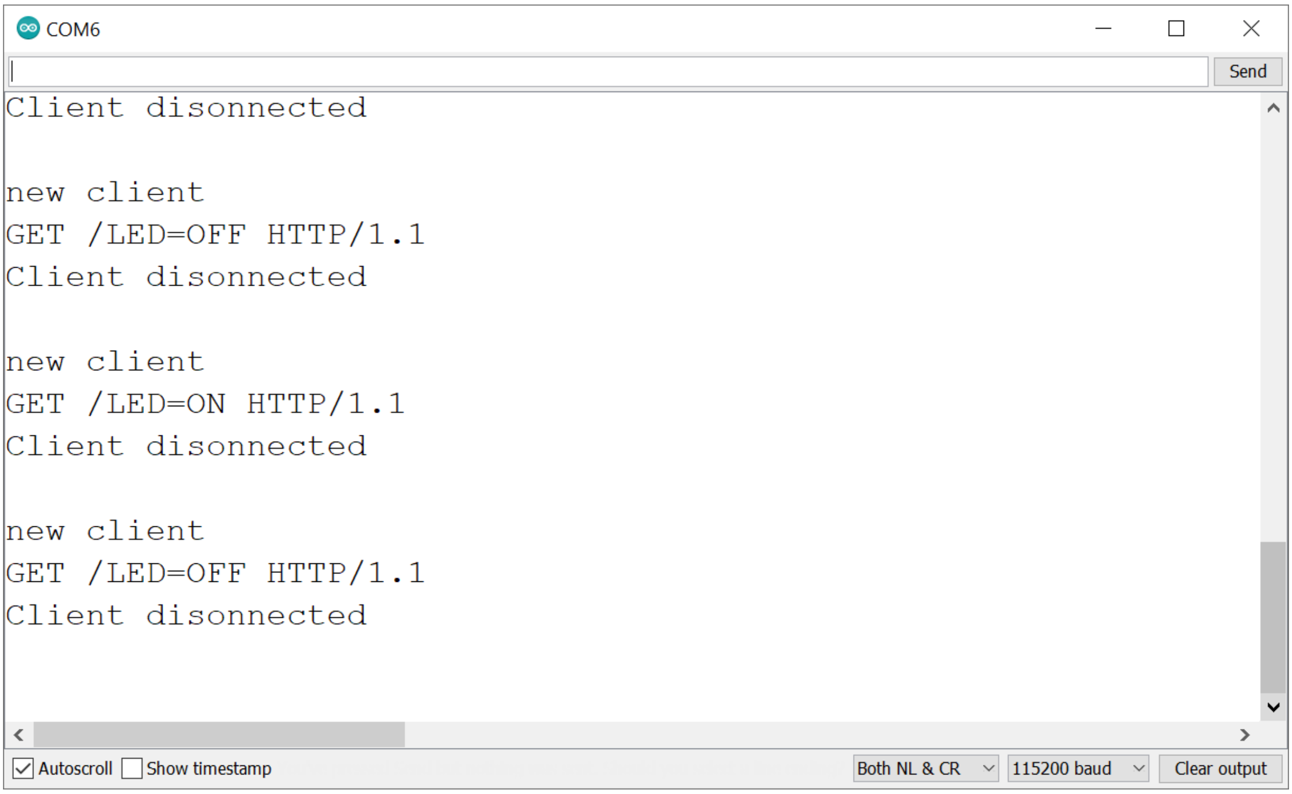


រូបភាព ៣.៨៣ ៖ Serial monitor បង្ហាញថាមានគេតភ្ជាប់ server

ដើម្បីបើក LED ចុចលើប៊ូតុង On ពេលនោះ Led is now: Off និងប្តូរទៅជា Led is now: On ហើយអំពូល LED នឹងភ្លឺ ដូចបង្ហាញរូបភាព ៣.៨៤។ នៅពេលចុចលើប៊ូតុង Off ពេលនោះ Led is now: On និង ប្តូរទៅជា Led is now: Off ហើយអំពូល LED នឹងឈប់ភ្លឺ។



រូបភាព ៣.៨៤ ៖ ទំព័រនៅលើ web browser ក្រោយពេលចុចលើប៊ូតុង On



រូបភាព ៣.៨៥ ៖ Serial monitor របស់ NodeMCU បង្ហាញពីការចុចប៊ូតុង On និង Off

នៅពេលចុចប៊ូតុង On ឬ Off នៅក្នុង web browser នៅក្នុង Serial Monitor ក៏បង្ហាញស្ថានភាពឆ្លើយតបរបស់ server ដូចបង្ហាញក្នុងរូបភាព ៣.៨៥។

៣.៦.៨ ការបញ្ជា NodeCMU ដោយប្រើប្រាស់ Blynk App

Blynk ជា App ប្រភេទ Internet of Thinks (IoT) ដែលអាចប្រើប្រាស់សម្រាប់បញ្ជា និងបង្ហាញទិន្នន័យពី NodeMCU ពីចម្ងាយតាមអ៊ីនធ័រណេត។ គេអាចកែសម្រួល interface របស់ App អាស្រ័យទៅនឹងតម្រូវការប្រើប្រាស់ជាក់ស្តែង អាស្រ័យលើឧបករណ៍ដែលតភ្ជាប់ជាមួយវា។ ឧបករណ៍ប្រើប្រាស់ Wifi, Bluetooth, Ethernet, និង Serial Port អាចតភ្ជាប់ទៅ Blynk cloud ឬក៏ server នៅក្នុង local area network។

នៅក្នុងឧទាហរណ៍ខាងក្រោមនេះ គេនឹងប្រើប្រាស់ Blynk App ដើម្បីបញ្ជា LED ដែលតភ្ជាប់ទៅនឹង nodeMCU (ESP8266)។

ឧបករណ៍ដែលត្រូវការមាន ៖

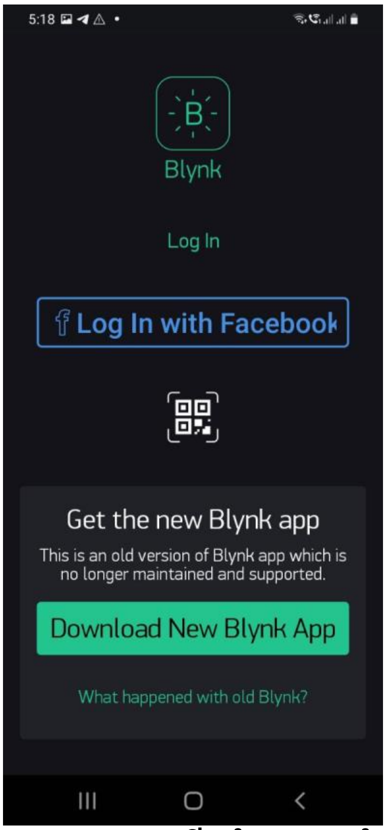
1. ESP8266 (NodeMCU)
2. LED

Software ដែលត្រូវការមាន ៖

Arduino IDE (1.8.5 ឬ ថ្មីជាងនេះ)

WiFi (លេខសម្ងាត់ credentials)

ដំឡើង Blynk App នៅលើ Smartphone។ នៅពេលបើកដំណើរការ វានឹងបង្ហាញស្រដៀងរូបភាព ៣.៨៦។

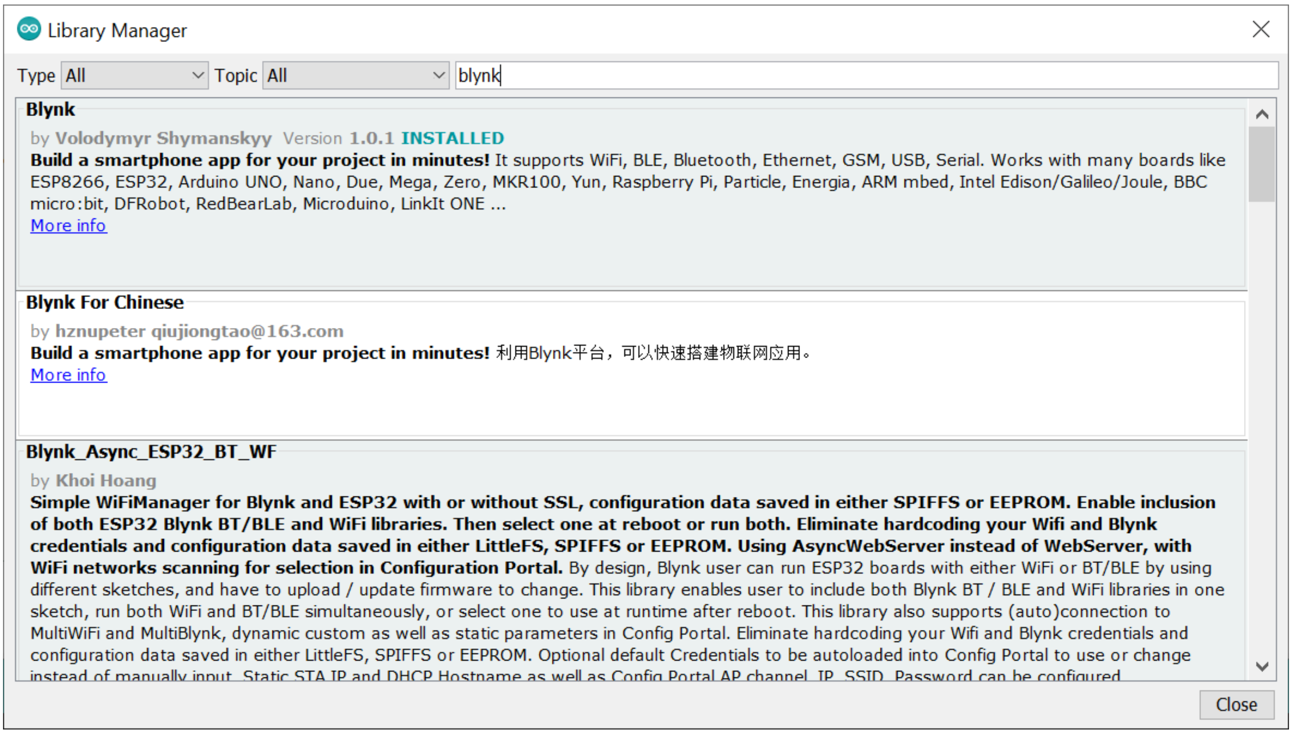


រូបភាព ៣.៨៦ ៖ Blynk app នៅពេលចាប់ផ្ដើមដំណើរការដំបូង

ដើម្បី login អាចចុចលើ Log In ឬ Log In with Facebook។

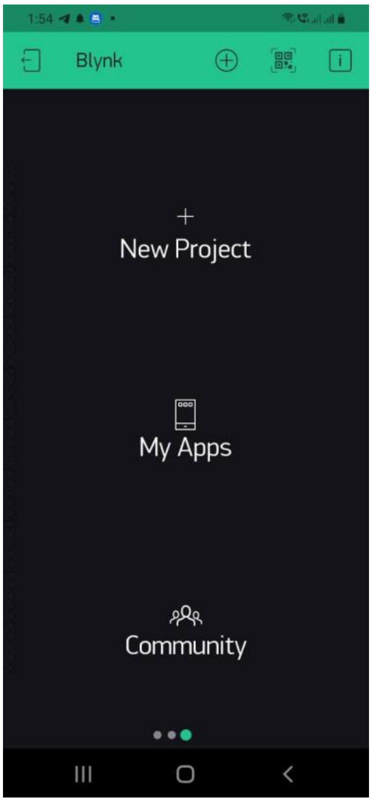
ដើម្បីបន្ថែម Blynk library ទៅក្នុង Arduino IDE សូមធ្វើដូចខាងក្រោមនេះ ៖

១. ចុច Sketch -> Include Library -> Manage Libraries ពេលនោះ Library Manager dialog box នឹងបង្ហាញដូចក្នុងរូបភាព ៣.៨៧។ នៅក្នុង text box វាយ blynk វានឹងបង្ហាញ Blynk library នៅផ្នែកខាងក្រោម។ ចុច Install ដើម្បីបន្ថែមវាទៅក្នុង Arduino IDE។ ក្រោយពីវា download នឹង install រួចហើយ គេអាចចុចប៊ូតុង Close នៅផ្នែកខាងក្រោមខាងស្ដាំ ដើម្បីបិទ dialog box និងត្រលប់ទៅកាន់ Arduino IDE វិញ។



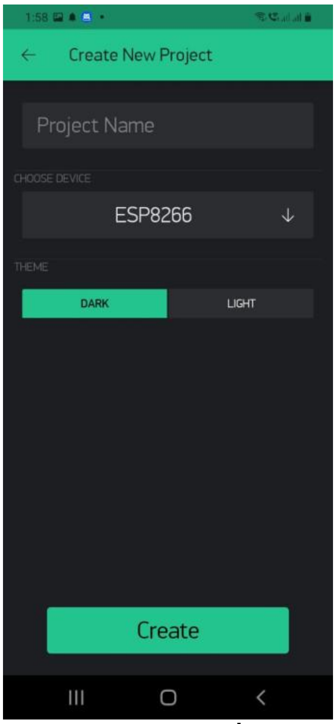
រូបភាព ៣.៨៧ ៖ Install Blynk library

មុននឹងបង្កើត Project ត្រូវបង្កើត account ឬ sign in ជាមុនសិន នៅក្នុង App។ នៅពេល sign in រួចហើយ វានឹងបង្ហាញដូចក្នុងរូបភាព ៣.៨៨ ខាងក្រោម។



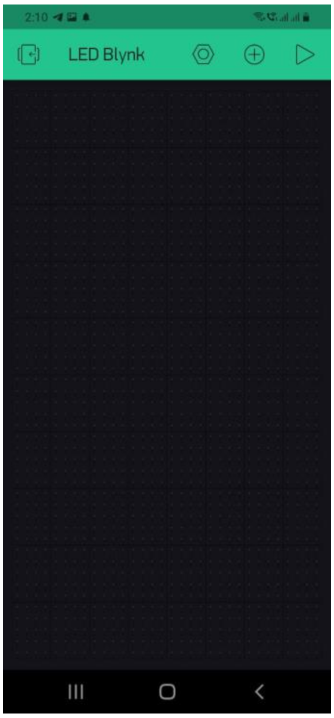
រូបភាព ៣.៨៨ ៖ Blynk App ក្រោយពេល log in

1. ចុចបង្កើត New Project ឬ ចុចលើ + នៅផ្នែកខាងលើ វានឹងបង្ហាញដូចក្នុងរូបភាព ៣.៨៩។



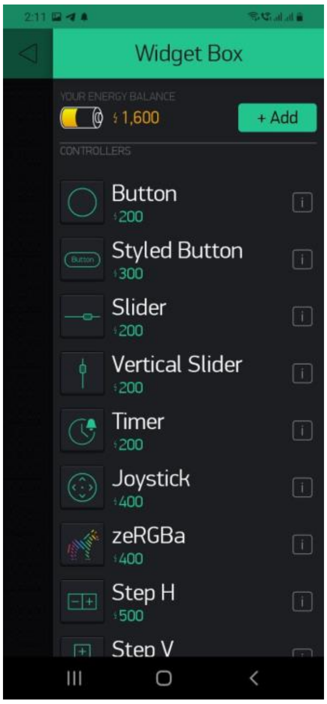
រូបភាព ៣.៨៩ ៖ ការជ្រើសរើសប្រភេទ NodeMCU

2. ត្រង់ Project Name វាយឈ្មោះគម្រោងថ្មី (ឧទាហរណ៍ LED Blynk) បន្ទាប់មក ជ្រើសរើសឧបករណ៍ (ឧទាហរណ៍ NodeMCU) វានឹងបង្ហាញឲ្យជ្រើសរើសប្រភេទតភ្ជាប់ (ឧទាហរណ៍ WiFi) រួចហើយចុចលើ Create។
3. វានឹងផ្ញើ Auth Token ទៅអ៊ីមែលរបស់យើង ចុចលើ OK វានឹងបង្ហាញរូបភាព ៣.៩០។



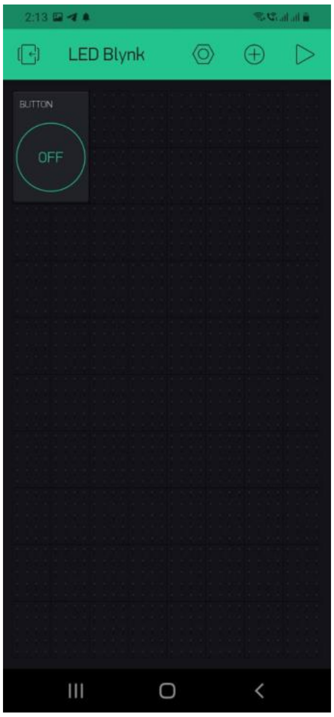
រូបភាព ៣.៩០ ៖ Blynk App ក្រោយបង្កើត New Project

4. បើក Widget Box (+) វានឹងបង្ហាញដូចក្នុងរូបភាព ៣.៩១។



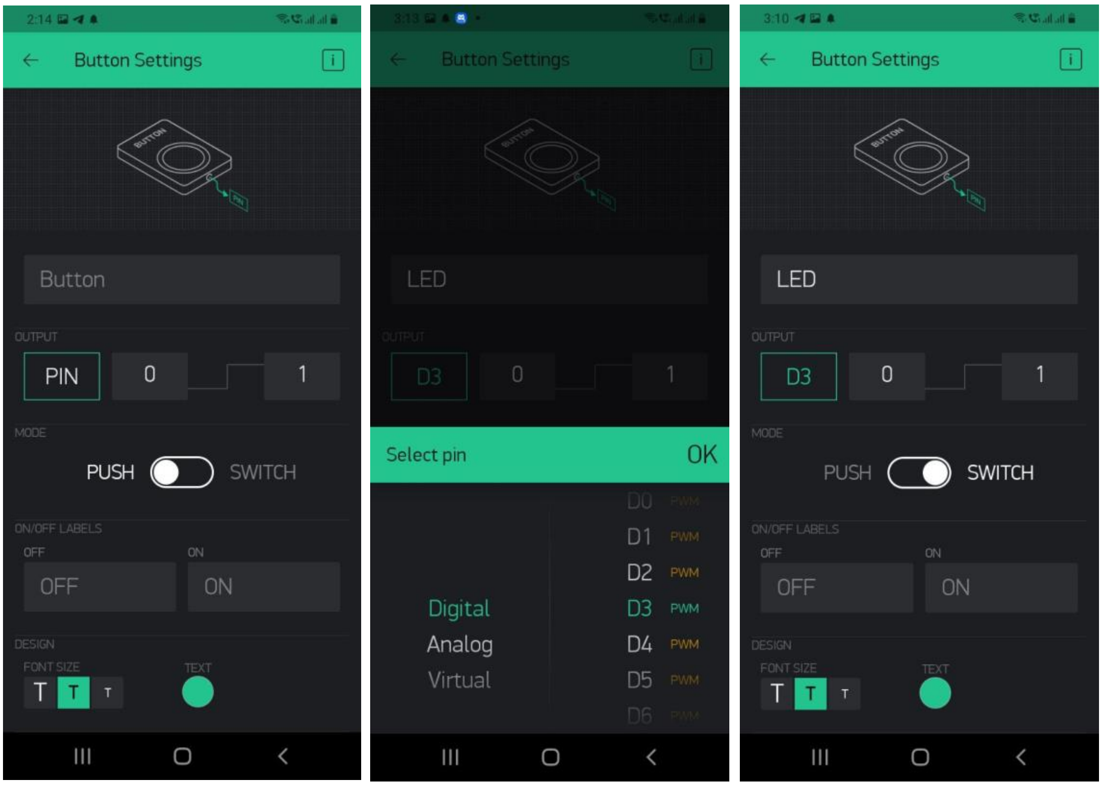
រូបភាព ៣.៩១ ៖ Blynk App ក្រោយពេលចុចបន្ថែម Widget Box

5. បន្ថែម Button ដោយចុចលើ Button។ វានឹងបង្ហាញដូចក្នុងរូបភាព ៣.៩២។



រូបភាព ៣.៩២ ៖ Blynk App ក្រោយពេលបន្ថែមប៊ូតុង

6. ដាក់ឈ្មោះឲ្យវានៅកន្លែង Button និង ចុចលើ PIN ដើម្បីជ្រើសរើស pin ដែល LED តភ្ជាប់ទៅនឹង NodeMCU (ឧទាហរណ៍ D3) រួចចុច OK រួចហើយជ្រើសរើសប្រភេទប៊ូតុងជា Switch។



រូបភាព ៣.៩៣ ៖ ការកែសម្រួលប៊ូតុងបញ្ជា LED

រួចចុច <- ដើម្បីត្រលប់ទៅវិញ។

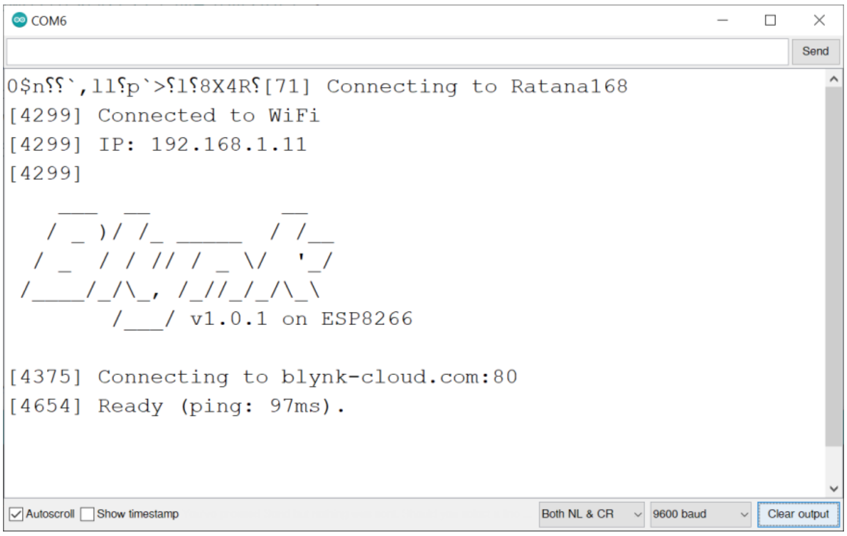
7. តភ្ជាប់អាណូតរបស់ LED ទៅ D3 និង កាតូតទៅ GND។

កូដ

កូដពីខាង client សម្រាប់បញ្ជា LED មានភាពសាមញ្ញ។

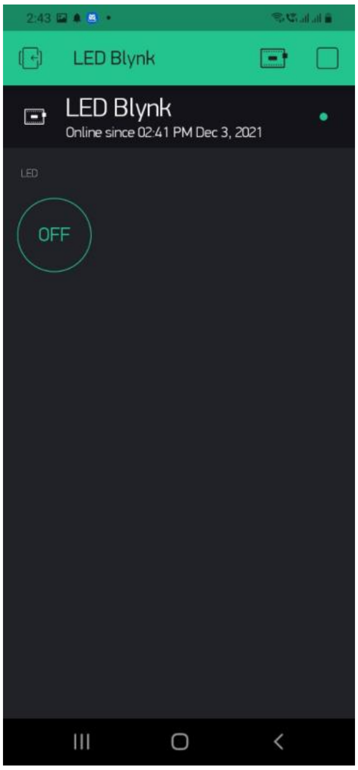
បើក Arduino IDE

1. ចុច File -> Examples -> Blynk -> Boards_WiFi -> NodeMCU (ជាប្រភេទ board ដែលប្រើប្រាស់)
2. វាយបញ្ចូល Auth Token ដែលផ្ញើមក នៅកន្លែង char auth[]
3. វាយបញ្ចូលឈ្មោះ WiFi និង password សម្រាប់ WiFi (char ssid[], char pass[])
4. Compile និង Upload កូដទៅក្នុង NodeMCU
5. បើក Serial Monitor និងពិនិត្យមើលទៅ NodeMCU បានតភ្ជាប់ទៅនឹង WiFi បានជោគជ័យដែរឬទេ? បើជោគជ័យ វានឹងបង្ហាញដូចក្នុងរូបភាព ៣.៩៤។



រូបភាព ៣.៩៤ ៖ Serial Monitor ពេល NodeMCU តភ្ជាប់ទៅ Blynk cloud បានជោគជ័យ

6. នៅលើ Blynk App ចុចលើ នៅផ្នែកខាងលើខាងស្តាំ ដើម្បីដំណើរការវា។ ចុចលើប៊ូតុង On ឬ Off ដើម្បីបើក ឬ បិទ LED ដូចបង្ហាញក្នុងរូបភាព ៣.៩៥។



រូបភាព ៣.៩៥ ៖ Blynk App បញ្ជា LED ពេលដំណើរការ

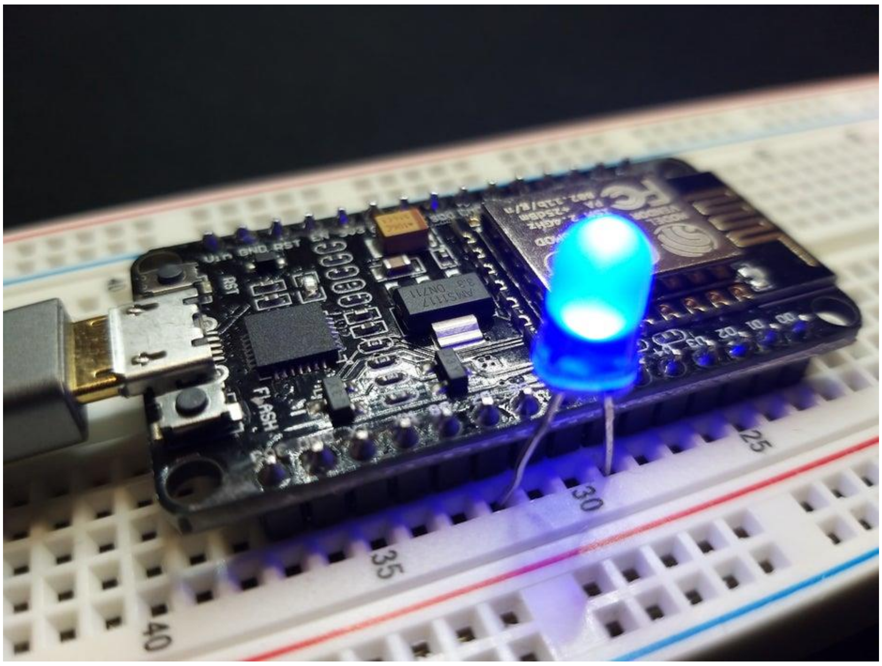
```
/**************************************************************

  Download latest Blynk library here:

    https://github.com/blynkkk/blynk-library/releases/latest


  Blynk is a platform with iOS and Android apps to control

  Arduino, Raspberry Pi and the likes over the Internet.

  You can easily build graphic interfaces for all your

  projects by simply dragging and dropping widgets.


    Downloads, docs, tutorials: http://www.blynk.cc

    Sketch generator:           http://examples.blynk.cc

    Blynk community:            http://community.blynk.cc

    Follow us:                  http://www.fb.com/blynkapp

                                http://twitter.com/blynk_app


  Blynk library is licensed under MIT license

  This example code is in public domain.
```

```
*************************************************************

This example runs directly on NodeMCU.


Note: This requires ESP8266 support package:

  https://github.com/esp8266/Arduino


Please be sure to select the right NodeMCU module

in the Tools -> Board menu!


For advanced settings please follow ESP examples :

 - ESP8266_Standalone_Manual_IP.ino

 - ESP8266_Standalone_SmartConfig.ino

 - ESP8266_Standalone_SSL.ino


Change WiFi ssid, pass, and Blynk auth token to run :)

Feel free to apply it to any other example. It's simple!

*************************************************************/


/* Comment this out to disable prints and save space */

#define BLYNK_PRINT Serial
```

```
/* Fill-in your Template ID (only if using Blynk.Cloud) */
//#define BLYNK_TEMPLATE_ID   "YourTemplateID"



#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup( )
{
  // Debug console
```

```
  Serial.begin(9600);



  Blynk.begin(auth, ssid, pass);

  // You can also specify server:

  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);

  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

}



void loop()

{

  Blynk.run();

}
```

ប្រសិនបើធ្វើបានជោគជ័យ គេអាចចុចបើកបិទ LED ពីចម្ងាយតាមអ៊ីនធ័រណេតដោយប្រើប្រាស់ Blynk App។



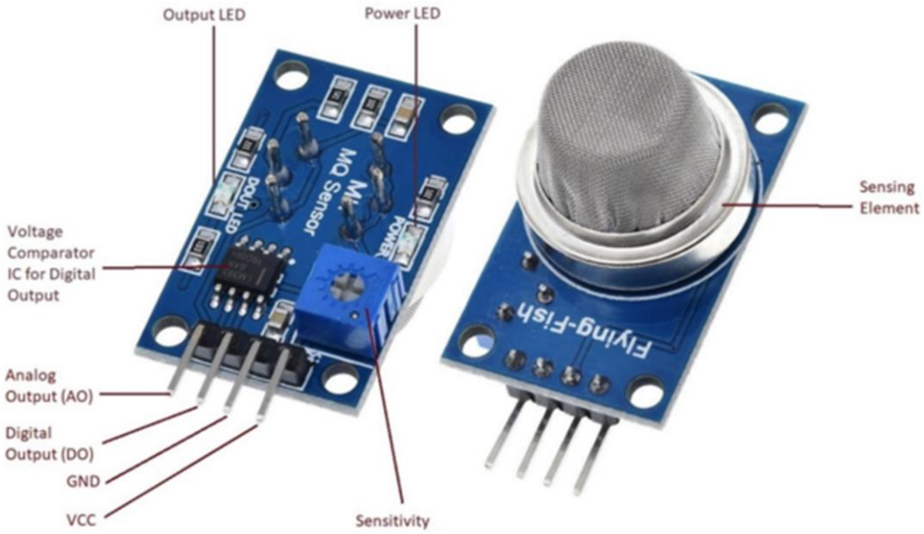
រូបភាព ៣.៩៦ ៖ ការតភ្ជាប់ LED ទៅ NodeMCU ដើម្បីអាចបញ្ជាពី Blynk App

៣.៧ ការវាស់វែងឧស្ម័នដោយប្រើប្រាស់ Arduino

ដើម្បីដឹងពីវត្តមាន ឬ កម្រិត នៃឧស្ម័នប្រភេទផ្សេងៗ គេអាចប្រើប្រាស់ប្រភេទសេនស័រមួយចំនួនតភ្ជាប់ទៅនឹង microcontroller ។ ឧទាហរណ៍ក្នុងវិស័យកសិកម្ម កាកសំណល់ដូចជា ទឹកកខ្វក់ពីកសិដ្ឋានចិញ្ចឹមសត្វ រោងចក្រកែច្នៃផលិតផលកសិកម្ម អាចប្រើប្រាស់សម្រាប់ផលិតជាជីវឧស្ម័ន ដែលក្នុងនោះមានឧស្ម័នម៉េតានដែលអាចប្រើប្រាស់ជាឥន្ធនៈសម្រាប់ដុតផលិតកម្តៅ និងអគ្គីសនី។ ក្នុងការប្រើប្រាស់ គេចង់ដឹងថា តើកម្រិតម៉េតាននៅក្នុងជីវឧស្ម័នមានកម្រិតណាដែរ? ចំណែកឯការចង់ដឹងពីវត្តមានឧស្ម័ននេះវិញ មានសារៈសំខាន់ណាស់ក្នុងករណីលិចជ្រាប ដើម្បីទប់ស្កាត់អគ្គីភ័យ និងការផ្ទុះ។

៣.៧.១ សេនស័រម៉េតាន

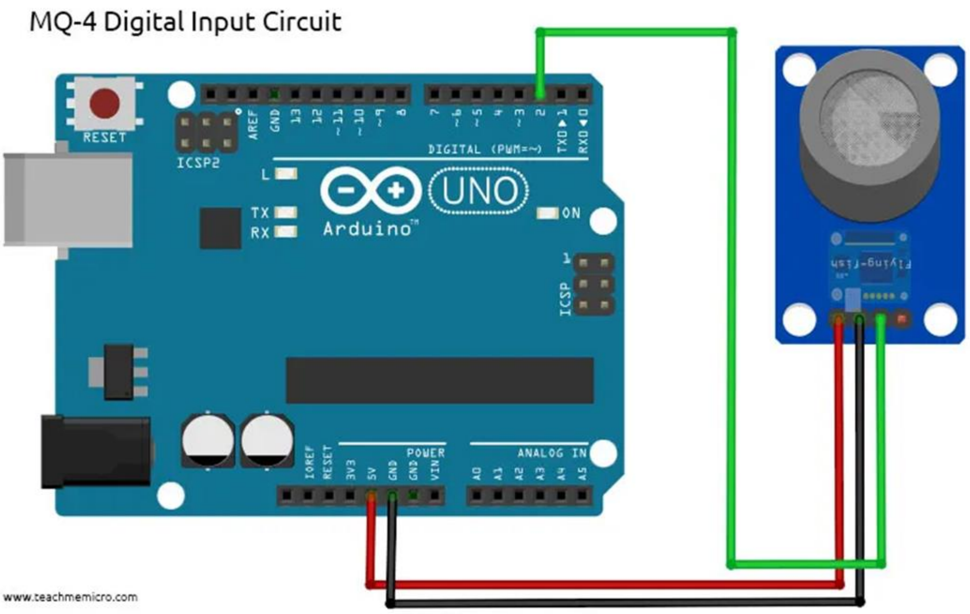
វាអាចវាស់ដឹងពីវត្តមាន និងកម្រិតម៉េតាននៅក្នុងជីវឧស្ម័ននៅក្នុងឡជីវឧស្ម័ន ឬកាយចេញពីទឹកស្អុយជាដើម។ ដើម្បីសម្រួលដល់ការប្រើប្រាស់ជាមួយ Arduino គេផលិតវាជា module ដូចក្នុងរូបភាព ៣.៩៧។



រូបភាព ៣.៩៧ ៖ Module សេនស័រម៉េតានសម្រាប់ប្រើប្រាស់ជាមួយ Ardunio

ប្រភព ៖ https://www.teachmemicro.com/mq-4-methane-gas-sensor-arduino/

ក្នុងករណីប្រើប្រាស់ដើម្បីដឹងពីវត្តមានម៉េតាន គេត្រូវប្រើប្រាស់ Digital Output pin ដូចក្នុងរូបភាព ៣.៩៨។

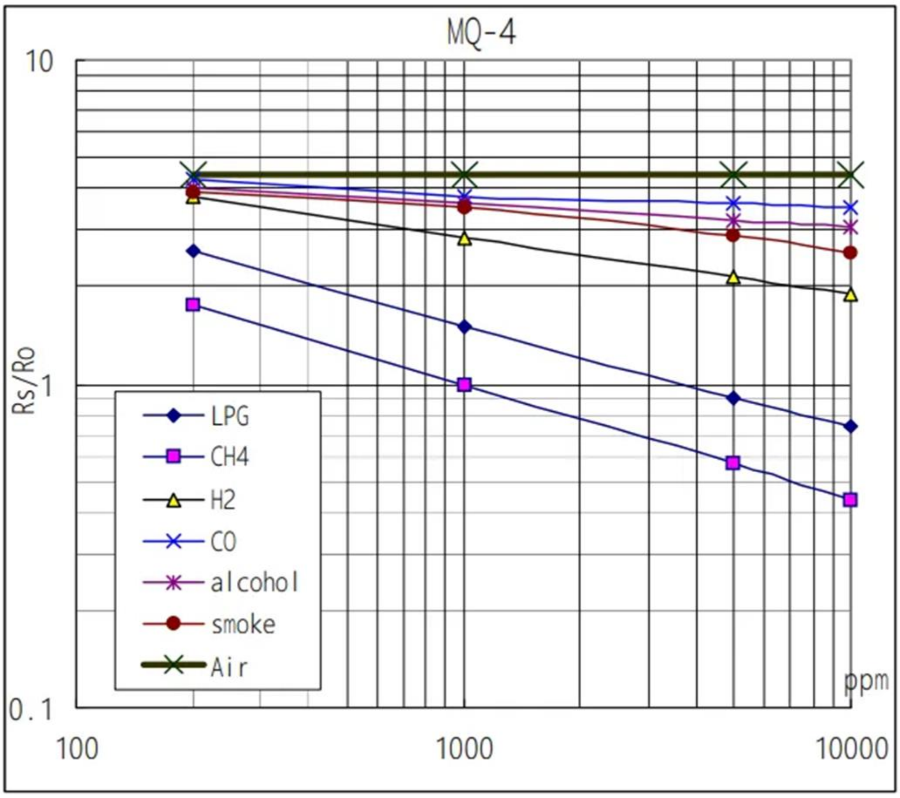


រូបភាព ៣.៩៨៖ ការតភ្ជាប់ MQ-4 សេនស័រ module ដើម្បីដឹងពីវត្តមានម៉េតាន

ប្រភព ៖ https://www.teachmemicro.com

ខាងក្រោមនេះជាកូដ

```
/* MQ4 Sensor - Digital Output Example
 * by R. Pelayo
 *
 * From TeachMeMicro (www.teachmemicro.com/arduino-mq4-methane-sensor
 *
 * Date Created: 09/11/2020
 */
const byte MQ4_Pin = 2; //MQ4 D0 pin

void setup() {
  pinMode(MQ4_Pin, INPUT_PULLUP);
   attachInterrupt(digitalPinToInterrupt(MQ4_Pin), sensor_triggered, CHANGE); //attach
interrupt on MQ4 pin
  Serial.begin(9600);
}

void loop() {
// Do anything you want here
}

void sensor_triggered() {
  Serial.println("Methane detected!"); // Output to serial monitor
}
```



រូបភាព ៣.៩៩ ៖ ទំនាក់ទំនងរវាង Rs/R0 និងកម្រិតឧស្ម័នប្រភេទផ្សេងៗ

ក្នុងការវាស់វែងកម្រិតម៉េតាន គេត្រូវដឹងពីទំនាក់ទំនងរវាងកម្រិតឧស្ម័ន និងផលធៀប RS រ៉េស៊ីស្តង់ ពេលវាស់ម៉េតាន ទៅនឹង R0 ពេលវាស់ខ្យល់ដូចបង្ហាញក្នុងរូបភាព ៣.៩៩។ ក្នុងករណីនេះ គេប្រើប្រាស់ Digital Output pin (A0)។

```
/* MQ4 Sensor - Analog Output Example
 * Prints out methane concentration in PPM to serial monitor
 * by R. Pelayo
 *
 * From TeachMeMicro ( www.teachmemicro.com/arduino-mq4-methane-sensor
 *
 * Date Created: 09/11/2020
 */
const byte MQ4_Pin = A0; //MQ4 A0 pin
const int R_0 = 945; //Change this to your own R0 measurements
```

```
void setup( ) {
  Serial.begin( 9600 );
}
void loop( ) {
  Serial.println( getMethanePPM( ) );
}

/*
 * getMethanePPM returns a float value in PPM of methane concentration
 */
float getMethanePPM( ){
  float a0 = analogRead( A0 ); // get raw reading from sensor
  float v_o = a0  * 5 / 1023; // convert reading to volts
  float R_S = ( 5-v_o )  * 1000 / v_o; // apply formula for getting RS
  float PPM = pow( R_S/R_0,-2.95 )  * 1000; //apply formula for getting PPM
  return PPM; // return PPM value to calling function
}
```

const byte MQ4_Pin = A0; //MQ4 A0 pin

សេនស័រ MQ4 តទៅនឹង pin A0។

const int R_0 = 945; //Change this to your own R0 measurements

តម្លៃរ៉េស៊ីស្តង់ R0 ប្តូរតម្លៃទៅតាមការវាស់រ៉េស៊ីស្តង់ជាក់ស្តែង។

Serial.println(getMethanePPM( ));

ព្រីនកម្រិតម៉េតានដែលវាស់ដោយសេនស័រ រួចចុះបន្ទាត់។

កូដក្នុង function getMethanePPM( ) ដំណើរការដូចខាងក្រោម។

float a0 = analogRead(A0); // get raw reading from sensor

អានតម្លៃ a0 ជាលេខមានក្បៀស ពីសេនស័រដែលតភ្ជាប់ទៅនឹង pin A0

float v_o = a0 * 5 / 1023; // convert reading to volts

បំលែងតម្លៃ a0 ទៅជាវ៉ុល រួចហើយទុកតម្លៃរបស់វានៅក្នុងអញ្ញត្តិ v_o ដោយគុណតម្លៃ a0 នឹង 5/1023។

float R_S = (5-v_o) * 1000 / v_o; // apply formula for getting RS

គណនាតម្លៃ R_S ដោយប្រើរូបមន្ត (5 – v_o) *1000/v_o។

float PPM = pow(R_S/R_0,-2.95) * 1000; //apply formula for getting PPM

គណនាតម្លៃ PPM ដោយប្រើប្រាស់រូបមន្ត។

return PPM; // return PPM value to calling function

បញ្ចេញតម្លៃ PPM នៅពេលគេប្រើប្រាស់ function getMethanePPM( )។

បណ្ណាល័យសាស្ត្រ

Books:

Childs, P. R. N. (2014). Mechanical Design Engineering Handbook. Butterworth-Heinemann. Oxford, UK.

Frenzel Jr., L.E. (2010). Electronics Explained. Newnes. ISBN 978-1-85617-700-9.

Purdum, J. (2012). Beginning C for Arduino: Learn C Programming for the Arduino and Compatible Microcontrollers. ISBN-13 (pbk): 978-1-4302-4776-0 or ISBN-13 (electronic): 978-1-4302-4777-7.

Smith, D.W., Sims, B.G., O'Neil, D.H. (1994). Testing and Evaluation of Agricultural Machinery and Equipment: Principles and Practices. FAO. ISBN 92-5-103458-3.

Websites:

Arduino data logger using SD card and DHT11 sensor. Available at https://simple-circuit.com/arduino-sd-card-dht11-data-logger/

Arduino Sensors. Available at Zuyd University of Applied Sciences.

Arduino Uno Pinout. Available at https://www.theengineeringprojects.com

Basic Electronics. Available at http://engineering.nyu.edu/gk12/amps-cbri/pdf/Basic%20Electronics.pdf

Circuit Simulator. Available at https://www.tinkercad.com

DHT11 and DHT22. Available at https://www.howtomechatronics.com

Electronics. Available at https://www.explainthatstuff.com

Fritzing. Available at https://fritzing.org

How to make a temperature-controlled fan using Arduino. Available at https://maker.pro/arduino/projects/how-to-make-a-temperature-controlled-fan-using-arduino

How to Use MQ-4 Methane Gas Sensor. Available at https://www.teachmemicro.com/mq-4-methane-gas-sensor-arduino/

Human-centered Design. Available at: https://www.netsolutions.com

Internet Controlled LED Using NodeMCU. Available at https://www.instructables.com/Internet-Controlled-LED-Using-NodeMCU/

NPN and PNP Transistors. Available at https://cdn11.bigcommerce.com/s-fccf0zqr3o/images/stencil/190x250/uploaded_images/shoptransmitter-npn-transistor-pnp-transistor.png?t=1544445403

Potentiometers. Available at https://www.instrumentationtoday.com

Transistor. Available at https://www.iamtechnical.com