

ពិភាក្សាល័យបច្ចេកទេសអនុមេត្តិតាមលោកស៊ីលស្វាយរៀន

REGIONAL POLYTECHNIC INSTITUTE TECHO SEN SVAY RIENG

ដែលបានបង្កើតឡើងដោយបច្ចុប្បន្ននិង

Microcontroller Programming

Arduino UNO

រៀបរៀងដោយលោក ល៊ែន ស៊ីលស្វាយ

២០២១

សាខាអ្នកជន

ខ្ញុំចាត់ យ៉ាន់ សុភក្រុង ជាគ្រូបង្រៀន ដំណាច់រួម និងត្រួតពិនិត្យ នៅវិទ្យាសាន ពហុបច្ចេកទេស ភូមិភាគ ពេជ្រ សែន ស្តាយរៀង បានរៀងរៀងនិងចងក្រងសៀវភៅនៃខេត្ត ឡើងដើម្បីជាដំនឹងយស្តារីដី និង សិស្ស- និស្សិត ដែលសិក្សាដើរ នៅទីតាំងគ្រួង អគ្គិសនី វិទ្យាសាស្ត្រ កំពុងខែ ទាំងអ្នកចេះខ្លះ និងអ្នកមិនចេះ សម្រាប់ជា មូលដ្ឋាន គួរឱ្យជាមួយយបទពិសោធន៍របស់អ្នក ដែលចង់សិក្សា សរស់រក្សាទុកដបញ្ហាដាមួយ គ្រឿងនៅទីតាំងគ្រួង អាចសែនយល់បានខ្លះទេ ដើម្បីយានទៅអនុវត្តន៍ជាយខ្ពស់និងបាន ហើយក្នុង សៀវភៅនៃខេត្ត របៀបដំឡើងកម្មវិធីសម្រាប់សរស់រក្សាទុកដបញ្ហា អនុវត្តសរស់រក្សាទុកដបញ្ហា របៀបយកក្នុង ដែលបានសរស់រក្សា ជាក់ចូលគ្រឿងនៅទីតាំងគ្រួង និងមានលំហាត់ពិសោធន៍ ដែលមានក្នុងគ្រឿង រួមជាស្របចំនួន ២២ ពិសោធន៍ ដោយប្រើប្រាស់ផ្ទាល់តាមលំហាត់រាយការអនុវត្តន៍និមួយ។ ដើម្បីជួយ អ្នកសិក្សាដាយតាមបញ្ហាបាន នូវចំណោះដើម្បីជួយគ្រឿង អាចធ្វើបាន និងជាក់ស្តែង ។

ខ្ញុំបានសង្ឃឹមថា សៀវភៅដែលបានចេញផ្សាយនេះ នឹងអាចធ្វើបំពេញនូវតម្រូវការរបស់អ្នកសិក្សា លោកគ្រូ អ្នកគ្រូ និងសិស្ស-និស្សិត ទាំងអស់ ខិតខ្សោយបែងសិក្សាស្រាវជ្រាវទៅលើការ សរស់រក្សាទុដ្ឋានមួយគ្រឹះដែឡើចត្រូនិច ដើម្បីតម្រូវទៅនឹងការកំណើនវិភាគនៃ នៃបច្ចេកវិទ្យាតួន្យសម្រាប់បច្ចេប្រឈមនេះ ។

ដោយសារពេលវេលា និងចំណោះយលប់ដីជាមានកម្រិត ម្ចាស់ហើយទោះបីជាមានការខិតខំយ៉ាងណាក់នៅមាន កំបុសខុសផ្ទាល់ដែលមានកើតឡើង ហើយ ខ្ញុំចាប់ស្ថាគមន់ដីកំភ្លាងល់ការវិន័យនិងទិនាន និងពីសំនួលមធរបស់ អស់ លោក លោកស្រីដើម្បីស្ថាបនាអឡើងវិញ្ញាងល់កង្វៈខាតដោយក្នុងវិករយក្រែកលេង លោកអ្នករាជទំនាក់ទំនងការប្រើយដ្ឋានអីមែល sopheakya123@gimail.comនៃ៖បានសូមអគ្គិន។

ថ្វីពុធ លោកស្រីត ខែមាយ ឆ្នាំ ជូន ព.ស.២៥៦៤

ស្តាយអេង ថ្ងៃទី ២០ខែ មករាឆ្នាំ២០២១

សេចក្តីថ្ងៃទី៣

មាតិកា

គម្រោងកថា	i
មាតិកា	ii
បញ្ជីអន្តរការតែ	vii
ចំណូន ១ ៖ ផ្ទល់ខ្លួនស្ថានស្រីសេដ្ឋិប្រចាំសិច (Basic Electronics)	1
មេធ្លើលទ្ធផល ១ ៖ ស្រីប្រចាំសិច	1
១. នសិស្សដៃ	1
២. កុងដឹង (Capacitor)	10
៣. ឌីយូត (Diode)	16
៤. ក្រពិសិស្សវ (Transistor)	19
៥. អំងុចតុង (Inductor)	20
មេធ្លើលទ្ធផល ២ ៖ បាត់តើរី (Battery)	21
៦. អីទៅជាបាត់តើរី	21
៧. សំណើលីច្ចេម	23
៨. ភាគុយ (សំណើលសំណើ-ភាសិត)	24
៩. សូឡាបែបិល	25
៥. ការពាក់ពេជាសើរី និងខ្លួន	26
១០. តារាងប្រភេទពពួកបាត់តើរី	27
១១. ប្រភេទនិមិត្តសញ្ញា	28
មេធ្លើលទ្ធផល ៣ ៖ ត្បូយសិទ (FUSE)	29
៦. អីទៅជាត្បូយសិទ	29
៧. សំណើនៃនំត្បូយសិទ	30
៨. បាត់តើរីដែលបានប្រឆាំងនៃត្បូយសិទ	30
៩. ទន្លេកំពង់ស្រួល	32
៥. ការចិត្តប្រើប្រាស់នៃគុណភាពដោយកត្តាសិតុណ្ឌភាព	32
៦. ត្បូយសិទយានយន្ត	33

ເຫດຜູ້ລະຈົບຂອງ ຂຸ່ມຄາສ (Switch)	34
១. ນິຍົມນໍຍ	34
២. ກາຣຄູກ	34
៣. Actuator	37
៤. ກຸ່ນຄາກໍເຜື່ອງວັດ	37
ບໍລິຫານ : ຕີເຫວັນລະເຫັນຕີແຈ້ງແຈ້ງປະເທດທີ່ໄດ້ຮັບຮັດ	40
ຕີເຫວັນ ១ : LDR	40
ຕີເຫວັນ ២ : ເລັກື່ງຮບສ່ວນ Motor DC	41
ຕີເຫວັນ ៣ : ກາຣກກໍເມື່ອແສົງສູ່ສູ່ໃໝ່ສ່ວນ LED	42
ຕີເຫວັນ ៤ : LED Blinking ເພີ້ມເປົ້າ IC555	43
ບໍລິຫານ : Microcontroller	44
ເຫດຜູ້ລະຈົບຂອງ Microcontrollers	44
១. ລັກຸດ: ໂັດ Microcontrollers ສືບໍ່ Microprocessors	44
២. ດາວໂຫຼວງໂຟຣ໌ Microcontroller	46
៣. ສົ່ງຕັ້ງລ່ວມ Reset	50
ເຫດຜູ້ລະຈົບ ២ : ຂວາມຕິດຂໍ້າລາ ລົບອະໂຄນະຊຸດຂະໜາດ: ອະລັດ Arduino UNO	51
១. ປົກລົງລັກຸດ: ຮບສ່ວນ Arduino UNO	51
២. ບຸກປະກາດຜູ້ຕິດ (POWER)	52
៣. ບຸກເພື່ອຊື່ຜິດລ (Digital Pin)	52
៤. ບຸກເພື່ອຄາດກາໂຄງ (Analog Pin)	53
៥. ຊໍານາກໍ່ຕິດ (Communication)	54
៦. ພັນຍັນຕຳ (Memory)	54
ເຫດຜູ້ລະຈົບ ៣: ຂວາມຈົບແຈ້ງອະນຸຍາກທີ່ຈີ ລົບການເປົ້າໃຈທີ່ຈີ	55
១. ກາຣັດເຊື້ອງກຸມວິຜິຕິ Arduino	55
២. ມຸຂ້າງກຸ່ນຜູ້ຕິດກຸມວິຜິຕິ Arduino	58
៣. ແບ່ງບັດກໍາຕິດຕິກຸມ Arduino ດ້ວຍກຸມທີ່ຈີ	59
៤. ກາຣເສສາກລົງຜູ້ຕິດຕິກຸມ Arduino ດ້ວຍກຸມທີ່ຈີ	61
ບໍລິຫານ ៤ : ໄຂນລົມອະນຸຍາກທີ່ຈີ ອະລັດເລຸ້ມື້ນູາ	64

ເຫດຜູ້ລາຍລືມ ຂາຍເນັດຂໍ້ຕົກລວມທີ່ສານາຄະດູໂທີ່ C	64
១. ປຽບແຕ່ກາສາ C	64
២. ກຸຽມວິທີເນັດກຸ່ນກາສາ C	64
៣. ງຳຮັ້ງເບບບອບສ່ວນກຸຽມວິທີກຸ່ນກາສາ C	66
ເຫດຜູ້ລາຍລືມ ២ : ຂະໜາວິທີ່ໄລສານາ C ສູ່ລາຍພາຊູ້ມື້ນູ້	69
១. ດາວໂຫຼວງເຕີມກຸຽມວິທີ	69
២. ກາຮຜົມບົງຍໍ (Comment).....	70
៣. ອົປະກິດໄລກາສາ C ກຸ່ນກາຊູ້ມື້ນູ້	70
៤. ປະເທດຕີ້ນຸ້ນໍ້ຍ (Data type)	72
៥ . ກາຣັດກໍລະບຸວະນາຍາ (Variable Name)	73
៦. ກາຣັດກາສາວະນາຍາ	74
៧. PIN direction	76
៨. ພັດທະນາກຸ່ນ	77
ເຫດຜູ້ລາຍລືມ ៣ : ຢູ່ທານາທີ່ໄລສານາ C ສູ່ລາຍພາຊູ້ມື້ນູ້	80
១. ຕໍ່ໂປ່ງເບີຣ (Constants).....	80
២. ປະເທດວິທີ (Operators)	84
ເຫດຜູ້ລາຍລືມ ៤ : ຂາຍກຸ້ມາຈົດຕືລືສູງເຫັນຫຼາຍແຕ່ງໆ (Control Statements)	92
១. ອົປະກິດ if control statements	92
២. ອົປະກິດ switch case control statements	94
៣. ອົປະກິດ While () Loops structure.....	96
៤. ອົປະກິດ do...While () Loops structure	97
៥. ອົປະກິດ goto Statement	98
៦. ອົປະກິດ for Statement.....	98
៧. ອົປະກິດ break Statement.....	100
៨. ອົປະກິດ continue statements.....	101
៩. ອົປະກິດ Function (return)	101
ເຫດຜູ້ລາຍລືມ ៥ : Serial Monitor and String Function	104
៩. Serial Monitor	104

၁၅. String Objects	115
ចំណែក និង Arduino Projects	124
ពិសោធន៍ទី ១ និង Digital Output	125
ពិសោធន៍ទី ២ និង LED Blinking	128
ពិសោធន៍ទី ៣ និង Light Chaaser	130
ពិសោធន៍ទី ៤ និង Digital Input	134
ពិសោធន៍ទី ៥ និង Tilt Sensor	139
ពិសោធន៍ទី ៦ និង Single Digit 7-segment	142
ពិសោធន៍ទី ៧ និង 2 Digit 7-segment make counter 00-99	146
ពិសោធន៍ទី ៨ និង Control LED using Serial Command	148
ពិសោធន៍ទី ៩ និង Temperature Sensor (LM35)	152
ពិសោធន៍ទី ១០ និង Light Dependent Resistor	154
ពិសោធន៍ទី ១១ និង LCD display	158
ពិសោធន៍ទី ១២ និង IR Remote	165
ពិសោធន៍ទី ១៣ និង Servo Motor	175
ពិសោធន៍ទី ១៤ និង DC Motor	179
ពិសោធន៍ទី ១៥ និង Step Motor	182
ពិសោធន៍ទី ១៦ និង Distance Sensor	183
ពិសោធន៍ទី ១៧ និង Gas Sensor	189
ពិសោធន៍ទី ១៨ និង KeyPad	190
ពិសោធន៍ទី ១៩ និង Stepper Motor	194
ពិសោធន៍ទី ២០ និង DHT11 Sensor	197
ពិសោធន៍ទី ២១ និង Alcohol (MQ-3) Sensor ជាមួយ LCD និងbuzzer	199
ពិសោធន៍ទី ២២ និង Interfacing HC-05 Bluetooth Module	202
ឯកសារយោបល់	211

បញ្ជីអក្សរកាត់

(List of Abbreviation Words)

ល.ន	អក្សរកាត់	អត្ថលេខាលោង
1	ASCII	American Standard Code for Information Interchange
2	ALU	Arithmetic and Logic Unit
3	A/D	Analog to Digital Converter
4	BCPL	Basic Combine Programming Language
5	C	Capacitor
6	Cds	Cadmium sulfur photocell
7	CTR	Critical Temperature Resistor Thermistor
8	CPU	Central Processing Unit
9	D	Diode
10	D/A	Digital to Analog Converter
11	DPDT	Double Pole Double Throws
12	EEPROM	Electrical Erasable Programmable Read Only Memory
13	EPROM	Erasable Programmable Read Only Memory
14	ESR	Equivalence Series Resistance
15	GND	Ground
16	I	Current
17	I/O	Input & Output
18	I ² C	Inter-integrated Circuit
19	IC	Integrated Circuit
20	LED	Light-Emitting Diode
21	LCD	Liquid-crystal display

22	LSB	Least significant bit
23	LDR	Light Dependent Resistor
24	MSB	most significant bit
25	NC	Normally closed
26	NO	Normally open
27	NTC	Negative Temperature Coefficient Thermistor
28	OTP	One Time Programmable
29	PWM	Pulse Width Modulation
30	PCB	Printed circuit board
31	PTC	Positive Temperature Coefficient Thermistor
32	R	Resistor
33	Rx	Receiver
34	RAM	Random Access Memory
35	ROM	Read Only Memory
36	RS-232C	Recommended Standard 232, Revision C
37	SMD	Surface-mount device
38	SPI	Serial Peripheral Interface
39	SPDT	Single Pole Double Throw
40	SPST	Single Pole Single Throw
41	Tx	Transmitter
42	USB	Universal Serial BUS
43	V	Voltage
44	WDT	Watch Dog Timer

ចំណែក ១ : ឯ៉ានខ្មោនត្រីធម៌និងបច្ចេកទេស

ថ្លើបើ Arduino ជាប្រកែទ Board ដែលមានសៀវភៅត្រូវបានដោឡូង ឯងចានដែរ ពីរបាយការណ៍តែង ឱ្យរក្សាលអេឡិចត្រូនិចមួយតែប៉ុណ្ណោះ ម៉ោះហើយវាគ្មោះថ្លើបើការដោមួយ សៀវភៅ អេឡិចត្រូនិចដីទៅទៀតដែលប្រែបងើចជាទីផើនឹងដើរបស់មនុស្សយើងដែរ ដើម្បីឲ្យប្រពិបត្តិការ នៃដំណើរការរបស់របាយការពេញលេញ។ នៅក្នុងជំពូកទី១នេះ សិក្សាតីមូលដ្ឋានត្រី៖ ដំបូងអំពីគ្រឿងអេឡិចត្រូនិច មានដូចជា ធម៌ស្ថិស្ថិ កង់ដង់ ឱិយ៉ុត ត្រង់ស្ថិស្ថិ និងធ្វើនឹងដែលគ្រូយកមកអនុវត្តន៍ ជាមួយ Arduino គ្រាន់តែបង្ហាញពីរបាយនិងការអានតំលៃសំខាន់របស់រានិងចរិកលក្ខណៈទេ។

ចំណែក ១ : ឯ៉ានខ្មោនត្រីធម៌និងបច្ចេកទេស

បណ្តាលគ្រឿងបរិភោគដែលបានដោឡូងដោយអេឡិចត្រូនិច ចំកចោរដាក់ពីរប្រកែទេ:

- ប្រកែទអសកម្ម៖ ធម៌ស្ថិស្ថិ កង់ដង់ ឬបីន ។
- ប្រកែទសកម្ម៖ ឱិយ៉ុត បណ្តាលប្រកែទត្រង់ស្ថិស្ថិ អំពូល....ប្រកែទគ្រឿងបរិភោគទាំងនេះ មានអំពើយ៉ាងសកម្មទៅលើស្ថិស្ថិ (បំលែងស្រើស្រាវ ឬធ្វើក...) ។

១. នៅឯ៉ានខ្មោន

ធម៌ស្ថិស្ថិ គឺជាដែប្រឿងបរិភោគអេឡិចត្រូនិចមួយប្រកែទដែលគោលរាល់យកមកប្រើប្រាស់យ៉ាងទុលំ ទុលាយ ដើម្បី យាត់យោងចរន្តអគ្គិសនី និងធ្វើជាមុខងារមួយចំនួនធ្វើនឹងទៀត ។

ឯកសារក្នុង



គ្រាន់តែបង្ហាញពីរបាយ នៅក្នុងសៀវភៅ នៅក្នុងសៀវភៅ ជាប្រើប្រាស់យ៉ាងទុលំ ទុលាយ ដើម្បី យាត់យោងចរន្តអគ្គិសនី និងធ្វើជាមុខងារមួយចំនួនធ្វើនឹងទៀត ។ តម្លៃស្ថិស្ថិគឺតិចជាអូម (Ω) ដែលក្នុងនោះ 1000Ω មានតំលៃស្រើនឹង 1 គីឡូអូម (1KΩ) ហើយ 1000 KΩ មានតំលៃស្រើនឹង 1មេហ្មាមូម (1 MΩ) ។

ធម៌ស្ថិស្ថិ ចំកចោរដាក់ពីរក្នុង គឺនៅឯ៉ានខ្មោនត្រីធម៌និងបច្ចេកទេស និងនៅឯ៉ានខ្មោនត្រីធម៌និងបច្ចេកទេស ។

១.១. នគរូបាយកំណើនទេស

នសិក្សដែលបានបង្កើតឡើងនូវការបាន ។



នគរបាល ១ ៖ ប្រធ័នសុខសូងចំពេជ្យចេរដៃលប្រើប្រាស់ពីកញ្ចប់

នសិស្សដែលបច្ចេកទេសការងារបានធ្វើឡើងពីសរុបសកាយនវក័សរុបលោហ៍។ មិនត្រឹមតែបុណ្ណោះ
មានប្រភេទឡើងឡើតទៅតាមការនិយមរបស់គោ។ តំលៃរបស់នសិស្សដែលធ្វើឱ្យបាន តែម្មយកឱ្យម្បី
កំណត់ការដ្ឋីសរុបនសិស្សដែលក្រោមឈរបានស្ថិតិនោះទេ បុន្ថែនទៅមានកំរឿតឡើងនិងអត្រាអនុ
ភាពដែលជាបញ្ហាសំខាន់របស់នសិស្សដែលធ្វើឱ្យបាន កំរឿតឡើងរបស់នសិស្សដែលបានកំពុងតំលៃរបស់នសិ
ស្សដែល និងតំលៃរបស់នសិស្សដែលមាននៅលើក្នុរបស់វា ឧទាហរណ៍ កំរឿតឡើង $\pm 5\%$
បញ្ហាកំចាត់តំលៃពីរបស់នសិស្សដែលធ្វើឱ្យបានជាងប្រុកប្រុងសំជាងតំលៃនៅលើក្នុរបស់វា 5% ។ អត្រាអនុភាព
បញ្ហាកំពុងតំលៃពីរបស់នសិស្សដែលធ្វើឱ្យបានក្នុងការប្រើប្រាស់។

អត្រាអានុភាពអគ្គិបរមាបស់នសីស្សដៃបញ្ចក់ទេវិធីដោយ វីត់ (W) ។ អានុភាពអាចតណ្ហាដោយប្រើប្រាស់ ការនៃចរន្តគុណនិងតំលៃនសីស្សដៃ ។ ប្រសិនបើអត្រាអានុភាពអគ្គិបរមានៃនសីស្សដៃ លើសវានិងធ្វើឡាក្នុងខ្លាំងហើយអាចបណ្តាលឲ្យផែះបាន ។ នៅក្នុងសៀវភៅអេឡិចត្រូនិចនសីស្សដៃ គឺមានអត្រាអានុភាព $1/8W$, $1/4W$ និង $1/2 W$ ។ $1/8W$ គឺប្រើប្រាស់ស្ថើរវំពេកប្រចាំសៀវភៅអេឡិចត្រូនិចដែលជាលំហេរសីព្វាល់ ។

តាំលេបស់នសិស្សដែលបានបង្ហាញប្រើប្រាស់ ក្នុងពណ៌ (ពណ៌ជាបាន វិទណ៌ជាឯួង) ពីក្រោម នសិស្សដែលបានបង្ហាញប្រើប្រាស់ ក្នុងពណ៌ (ពណ៌ជាបាន វិទណ៌ជាឯួង) ពីក្រោម

ក. ភាគអនត់ប្រជម្លេសទីស្តីស្តុចំ

២. ករណីនៃស្នើសាលាប់ខាង ៣ជំនួយនៃស្នើសាលាប់ ៤ជំនួយ

ចំពោះនសីស្សដែលបានបង្កើតឡើង និង បានបង្កើតឡើង ជាតិលេខខ្លួន និង បានបង្កើតឡើង ជាតិមេគុណ ចំពោះ បានបង្កើតឡើង ជាតិលេខបញ្ជីនរបស់នសីស្សដែលបានបង្កើតឡើង ។ ឧបន្យក្រាមគឺជាពាណាការងក្រឹងពណ៌របស់នសីស្សដែលបានបង្កើតឡើង ៥ ពណា

ລາຍລະ ១ ៖ ຕາກັນກູຜົດຕົວ ແ ຂັ້ນ

ពណ៌នសិស្ស	ខ្ពស់ទី១ ជាលេខ	ខ្ពស់ទី២ ជាលេខ	ខ្ពស់ទី៣ មេគុណ	កម្រិតលំអៀង
ខ្សោយ	-	0	x 1 Ω	-
ត្រូវតត្រូវ	1	1	x 10 Ω	± 1 %
ក្រហម	2	2	x 100 Ω	± 2 %
ទឹកក្រុច	3	3	x 1 K Ω	-
លើផ្តើម	4	4	x 10 K Ω	-
បែកដង	5	5	x 100 K Ω	-
ខ្សោរ	6	6	x 1 M Ω	-
ស្បាយ	7	7	x 10 M Ω	-
ប្រធែល់	8	8	-	-
ស	9	9	-	-
ប្រាក់	-	-	x 0.01 Ω	± 10 %
មាស	-	-	x 0.1 Ω	± 5 %

ଓঁ শশী পাত্র

នសិស្សដែលពាក្យម្បយ មានពណ៌ជាយ ខ្លួនទី១ គឺពណ៌ក្រហម ខ្លួនទី២ គឺពណ៌ខ្មែរ ខ្លួនទី៣ គឺពណ៌ក្រច្ច និងខ្លួនទី៤ គឺពណ៌មាស ។

ចំណើនីយេ គឺ ជាយ ឱ្យដឹងទិន្នន័យ ពីពលរដ្ឋប្រជាមុន គឺស្មើ 2

ឧង់ទី២ តីពណ៌ខ្មែរ តីស៊ី ០

ឧង់ទិោ គីឡូវិកក្រុច គីជាមេគុណ x 1 KΩ

និង ឱ្យដោឡើង គីតណុក មាស គីជាតា កម្រិតលំអេរ៉ីង គីស្ទឹង $\pm 5\%$

ឧត្តមស៊ី ២៩

នសិស្សដែលបានពារម្បយ មានព័ត៌មានថា ជាយ ខ្លួនទី១ គឺព័ត៌មានស្មាយ ខ្លួនទី២ គឺព័ត៌មានរំបៀប
ខ្លួនទី៣ គឺព័ត៌មានខ្លួនទី៤ និងខ្លួនទី៥ គឺព័ត៌មាន ប្រាក់។

ଜ୍ଞାନେତ୍ର କୀ ପୋଯି ଥାଏନ୍ତିର କୀଟଦର୍ଶକ କୀଣ୍ଵୀ ୭

ឧង់ទី២ គីឡូនីកបែតង គីសី ៥

ឧង់ទី៣ គីណុលាអ្វីវ គីជាមេគុណា $\times 1 M\Omega$

និង ខ្សែត្រួចតិចណា ប្រាក់ តិជា កម្រិតលំនៅង តីសី $\pm 10\%$

ដូចនេះ សរុបតម្លៃគ្រប់គ្រងទណ្ឌមួយ មានទណ្ឌា ជោយ ខ្លួនទី១ តើទណ្ឌស្តាយ ខ្លួនទី២
តើទណ្ឌបែកអង ខ្លួនទី៣ តើទណ្ឌខ្សោយ និងខ្លួនទី៤ តើទណ្ឌ ប្រាក់ តើមានតម្លៃ $75 \times 1 \text{ M} \text{ ល} = 75 \text{ M} \text{ ល}$
និង កម្រិតលំអោង $\pm 10\%$ ។

ક. કરબીએસ્ટેસ્ટુલ્યુલાન્ડ્સ પ્રોફેસિશન્સ ૫ ઇન્ડ્સ્ટ્રી

ចំពោះនស្សីស្សុង់ដ ពណ៌កី ខ្លង់ទី១, ខ្លង់ទី២ និង ខ្លង់ទី៣ ជាតាំលេលខ វិង ខ្លង់ទី៤ តីជាមេគ្ហោបាន ចំពោះខ្លង់ទី៥ តីជាតាំលេលរួម នស្សីស្សុង់នៅទៅ។ ឧងតារ តីជាតាការុងក្នុងពណ៌កីនស្សីស្សុង់ដ ពណ៌ក៖

តារាង ២ : តារាងក្នុងពណ៌ក ៥ខ្ពស់

ពណ៌កនូវឯកសារ	ខ្ពស់ទី១ ជាលេខ	ខ្ពស់ទី២ ជាលេខ	ខ្ពស់ទី៣ ជាលេខ	មេគូណា	កម្រិតលំអៀង
ខ្លួន	-	0	0	$\times 1 \Omega$	-
ត្រួត	1	1	1	$\times 10 \Omega$	$\pm 1 \%$
ក្រហម	2	2	2	$\times 100 \Omega$	$\pm 2 \%$
ទីក្រុង	3	3	3	$\times 1 K\Omega$	-
លើផ្ទឹង	4	4	4	$\times 10 K\Omega$	-
បែកនៃ	5	5	5	$\times 100 K\Omega$	$\pm 0.5 \%$
ខ្សោយ	6	6	6	$\times 1 M\Omega$	$\pm 0.25 \%$
ស្តាយ	7	7	7	$\times 10 M\Omega$	$\pm 0.1 \%$
ប្រធែះ	8	8	8	-	-
ស	9	9	9	-	-
ទីក្រាត់	-	-	-	$\times 0.01 \Omega$	$\pm 10 \%$
ទីកមាស	-	-	-	$\times 0.1 \Omega$	$\pm 5 \%$
ត្រានពណ៌ក	-	-	-	-	$\pm 20 \%$

ឧបាទាន់ ១:

នសិុស្សដែលត្រូវដោឡូលិច មានពណ៌ក ដោយ ខ្ពស់ទី១ តីពណ៌កក្រហម ខ្ពស់ទី២ តីពណ៌កខ្លួន ខ្ពស់ទី៣ តីពណ៌កទីក្រុង ខ្ពស់ទី៤ តីពណ៌ក ក្រហម និងខ្ពស់ទី៥ តីពណ៌ក មាស ។

ចំណេះ តី ដោយ ខ្ពស់ទី១ តីពណ៌កក្រហម តីស្ទើ 2

ខ្ពស់ទី២ តីពណ៌កខ្លួន តីស្ទើ 0

ខ្ពស់ទី៣ តីពណ៌កទីក្រុង តី 3

ខ្ពស់ទី៤ តីពណ៌ក ក្រហម តី ជាមេគូណា $\times 100 \Omega$

និង ខ្ពស់ទី៥ តីពណ៌ក មាស តីជាកម្រិតលំអៀង តីស្ទើ $\pm 5 \%$

ចូចចេះ: នសិុស្សដែលត្រូវដោឡូលិច មានពណ៌ក ដោយ ខ្ពស់ទី១ តីពណ៌កក្រហម ខ្ពស់ទី២ តីពណ៌កខ្លួន ខ្ពស់ទី៣ តីពណ៌កទីក្រុង ខ្ពស់ទី៤ តីពណ៌ក ក្រហម និងខ្ពស់ទី៥ តីពណ៌ក មាស តីមានតំលៃ $203 \times 100 \Omega = 20300 \Omega = 20.3 K\Omega$ និង កម្រិតលំអៀង $\pm 5 \%$ ។

ଓଡ଼ିଆ ପ୍ରକାଶନ

ବ୍ୟାକ୍ କୀ ପୋଯି ଦୁଇଟିଏକାଟି କୀଟଙ୍ଗାଣ୍ୟ କୀନ୍ତି 7

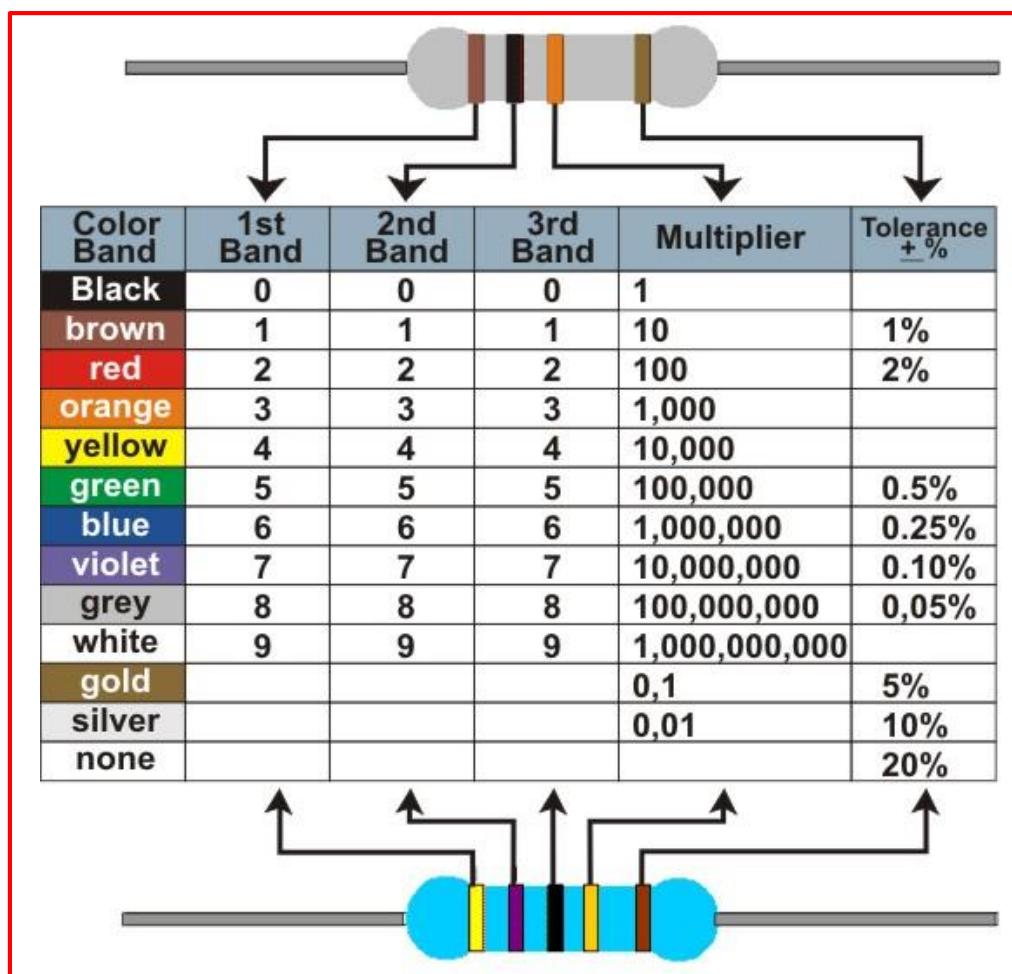
ឧង់ទី២ តីពណ៌បែបនេះ តីស្មើ ៥

ឧង់ទិន គីពណ៌ខ្មែរ គីសី 6

ឯង់ទី៤ គីឡូក្រុងតាមរបាយការណ៍ x 10 Ω

និង ឯងទី៤ តើពណ៌ ប្រាក់ តើជា កម្មិតលំអោង តើស្រើ $\pm 10\%$

ឧបនេះ ស្ថិតុដ្ឋីបូនពណ៌ម្មយ មានពណ៌ ដោយ ខ្លួនទី១ គិតពណ៌ស្សាយ ខ្លួនទី២ គិតពណ៌បែកនឹង ខ្លួនទី៣ គិតពណ៌ខ្សោយ ខ្លួនទី៤ គិតពណ៌ត្រាត និងខ្លួនទី៥ គិតពណ៌ ប្រាក់គិតមានតម្លៃ $756 \times 10 \Omega = 7560 \Omega = 7.56 \text{ K}\Omega$ និង កម្រិតលំអៅង $\pm 10\%$ ។



ຮຽນຮາຍ ២ ៖ កុដាបន្ទីរបស់រស្សីស្សាន់ និងខ្លួន

៤. ការអើយឱ្យស្នើសុំលក់ថ្លែង (SMD)

ពីមានពីរប្រភេទដែរសម្រាប់ការអារ៉ាតំលេបសៀវភៅ គីប្រភេទ ៣ តួលេខ មានកំរិតល្អូង ៥% និង ២% និង ប្រភេទ ៤ តួលេខ មានកំរិតល្អូង ១ % ។

❖ ចំពោះតាមលក់ថ្លែងប្រចេង ៣ តួលេខ

ប្រភេទកំរិតល្អូង ៥% និង ២%

R	និមិត្តសញ្ញា
1.....9.9	XRX
10.....99	XXR
100.....10M	XXX

ជូរទីបីជាចំណួនស្ថើយគុណនៃដំប់

ឧចាងន័ែ៖

$$4R7 = 4.7\Omega , 33R = 33 \Omega , 472 = 4.7 K, 153 = 15 K$$

❖ ចំពោះតាមលក់ថ្លែងប្រចេង ៤ តួលេខ

ប្រភេទកំរិតល្អូង ១ %

R	និមិត្តសញ្ញា
100 Ω.....999 Ω	XXXR
1K Ω	XXXX

លេខជូរទីបីនឹងជាស្ថើយគុណនៃដំប់

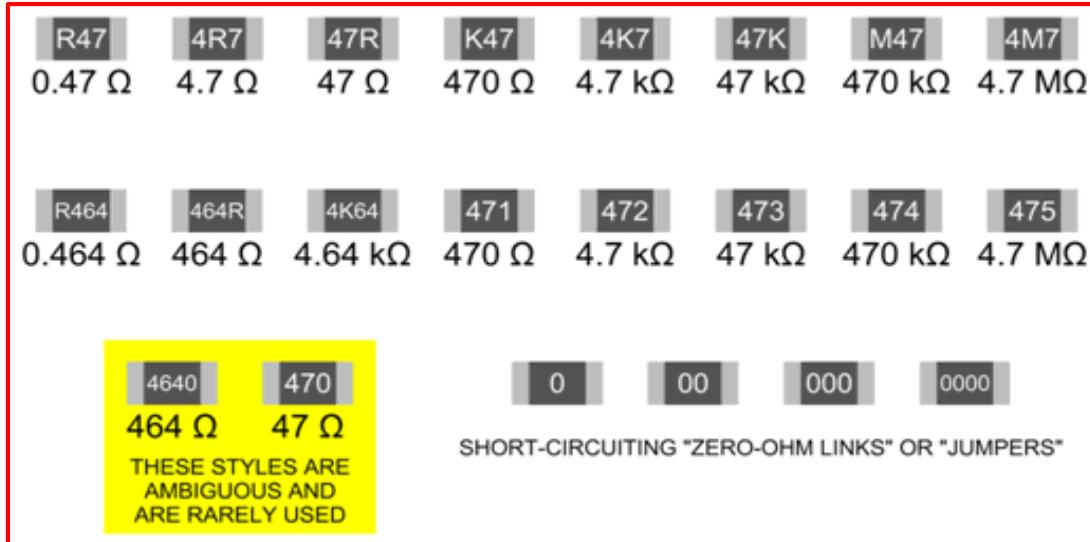
ឧចាងន័ែ៖

$$100R = 100 \Omega$$

$$330R = 330 \Omega$$

$$1502 = 150 00 \Omega = 15 K \Omega$$

$$1003 = 100 000 \Omega = 100 K \Omega$$



ຮູບລາຍງານ ຕິດຕະຫຼາດ ສົມມັກ SMD

១.២. នេតីស្សាគតំណែងប្រព័ន្ធមេន

កំរើយ៉ាបខ្លួន តិចជាប្រភេទកំរើយ៉ាបលីណោអីវិ ដែលតាំលែរបស់កាសមាមាត្រឡើងមំបង្កិល ។
កំរើយ៉ាបធ្លឹងមានជាប្រភេទកំរើយ៉ាបលីណោអីវិនិងមានជាប្រភេទក្រុបក្រុលតាំលែតាមអនុគមនីលោកវិត ។ បណ្តុាលតាំលែរបស់កំរើយ៉ាបធ្លឹងមាន៖ $100\ \Omega$, $220\ \Omega$, $470\ \Omega$, $1K\ \Omega$, $2.2K\ \Omega$, $4.7\ K\ \Omega$, $10K\ \Omega$, $20K\ \Omega$, $47K\ \Omega$, $100K\ \Omega$, $200K\ \Omega$, $470K\ \Omega$, $1M\ \Omega$, $2.2M\ \Omega$

បណ្តុាលតែលរបស់វិរីយ៉ាបខ្សោយរូមាន៖ $10\ \Omega$, $22\ \Omega$, $47\ \Omega$, $100\ \Omega$, $220\ \Omega$, $470\ \Omega$, $1K\ \Omega$, $2.2K\ \Omega$, $4.7K\ \Omega$, $10K\ \Omega$, $22K\ \Omega$, $47K\ \Omega$



រូបភាព ៤ : រូបភាពរសីស្សដែលបានរៀបចំ

ក. ផ្ទុកអសីស្សទៅ

ផ្ទុកអសីស្សជាការប្រើប្រាស់បានបង្កើតឡើងពីធាតុ Sulfur Cadmium ម្មានហើយនៅលើ និមិត្តសញ្ញា ប្រើប្រាស់មានសរស់អេក្រូវ Cds (Cadmium Sulfur Photocell)។ ផ្ទុកអសីស្សមានតម្លៃជំនួយ ៩ ដុល្លារ និងអារម្មណ៍សូត ពន្លឹះចំបៅះវា កាលណាពន្លឹះចំបៅះវានេះ នសីស្សមានតម្លៃ 200 អូម និងនៅពេលដឹកនោះ នសីស្សមានតម្លៃ ប្រាំហល ២ មេប្រាកម្ម។



រូបភាព ៥ : ផ្ទុកសីស្ស

ផ្ទុកអសីស្សជាបានប្រើយ៉ាងទូលំទូលាយក្នុងមុខងារ៖

- បិទបើកត្រីនឹងប័ណ្ណីផ្ទុវា
- ក្នុងតាក់ពន្លឹះ

២. ទែមីស្ទៃ (THERMISTOR –TH)

ទែមីស្ទៃគឺជាប្រភេទរៀងស្តីស្ថុដែលប្រប្រឈមតាំង អារ៉ាមីយោទាមសីតុណ្ឌភាព។ មានទែមីស្ទៃបី ប្រភេទ ៖

❖ NTC (Negative Temperature Coefficient Thermistor)

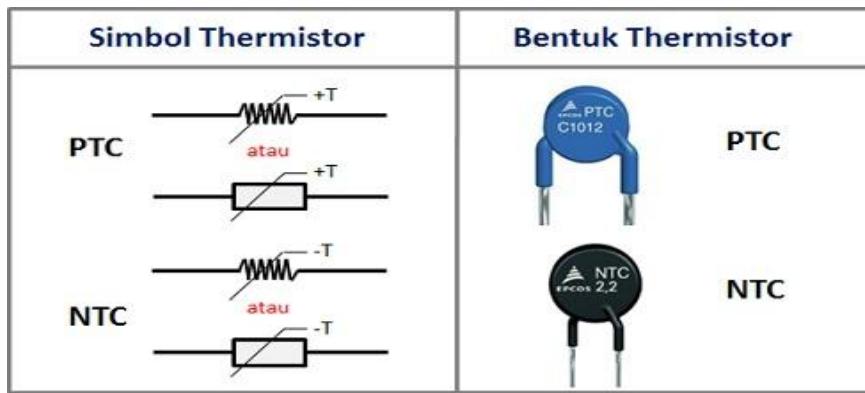
គឺជាទែមីស្ទៃដែលតាំងលើរៀងស្តីស្ថុដែលរបស់វាចិញ្ចួរបន្ទាប់នៅពេលសីតុណ្ឌភាព កើនឡើង ។ វាគ្រូរបានគេប្រើប្រាស់ដើម្បីត្រួតពិនិត្យសីតុណ្ឌភាព ។

❖ PTC (Positive Temperature Coefficient Thermistor)

គឺជាទែមីស្ទៃដែលតាំងលើរៀងស្តីស្ថុដែលរបស់វាកើនឡើងយ៉ាងឆាប់រហូតនៅពេល សីតុណ្ឌភាព កើនឡើងលើស ចំនួច ណាមួយដែលបានកំណត់ ។

❖ CTR (Critical Temperature Resistor Thermistor)

គឺជាទែមីស្ទៃដែលតាំងលើរៀងស្តីស្ថុដែលរបស់វាចិញ្ចួរបន្ទាប់នៅពេលសីតុណ្ឌភាព កើនឡើងលើស ចំនួចណាមួយដែលបានកំណត់ ។

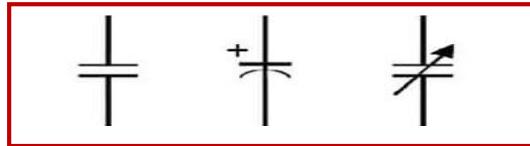


រូបភាព ៦ ៖ និមិត្តសញ្ញា និងរូបពិត THERMISTOR

៣. កុវប់ (Capacitor)

កុងដង (Capacitor or Condenser) គឺជាប្រព័ន្ធបង្កុងអេឡិចត្រូនិកអសកម្មដែលមាន ដើរពីរ ដែលមានក្នុងនាទីរក្សាទុកចាមពលក្នុងដែនអគ្គិសនីរបស់វាបាយណ៍រាជរាជ្យ ។ លក្ខណៈនៃកុងដង គ្រូរបានកំណត់ដោយចំនួនចែរមួយហេត្តា Capacitance ។ Capacitance គឺជាសមត្ថភាពនៃការផ្តើក បន្ទុកអគ្គិសនីនៃរូបធាងតុមួយ ។ រូបធាងតុដែលមាន Capacitance កាន់ តែមែននឹងយុទ្ធភាព រក្សាទុក បន្ទុកអគ្គិសនីកាន់តែប្រើប្រាស់ ។ Capacitance គឺជាដាក់តែ (F) ។

សិទ្ធិត្វួន្ទាន់



រូបភាព ៧ ៖ សិទ្ធិត្វួន្ទាន់នៃកុងដង់

ជាតុទេតំលៃរបស់កុងដង់(Capacitance) មានដូចជា ពីក្នុងធាតុ (pF) លាងណូណាតុ (nF) មិក្នុងធាតុ (μF)

បុរណបទ	អក្សរកាត់	មេគុណ	កម្រិតកាត់
ពីក្នុងធាតុ	pF	10^{-12}	0.000000000001F
លាងណូណាតុ	nF	10^{-9}	0.000000001F
មិក្នុងធាតុ	μF	10^{-6}	0.000001F
មិលីក្នុងធាតុ	mF	10^{-3}	0.001F
គីឡូក្នុងកាត់	kF	10^3	1000F

រូបភាព ៨ ៖ ខ្លាតមេគុណរបស់កុងដង់

២.១. ប៉ាន៊ីថែរូបតាមការបង្ហាញ



រូបភាព ៩ ៖ កុងដង់ប្រភាន់បូល (អេឡិចត្រូនិក)

- **ចំហៀក** : ចំហៀកសំដែរបែងប្រភេទខាងក្រោមនឹង Capacitance។ តែចាត់ទុកចាតាកុងដីដែល ត្រូវបានប្រភេទខាងក្រោមគួរបាន បើណាន Capacitance ដែល ត្រូវបានប្រភេទខាងក្រោមគួរបាន បើណាន Capacitance ដែល
- **Maximum Voltage** : គឺជាបន្ទាក់តង់ស្បែងអតិថិជនបានបង្កើតឡើងទាំងពីរនៃកុងដី។ ជាបន្ទាក់មាន ផ្ទុចជាតា 16V , 25V , 50V, 100V -ល- ។ ប្រសិនបើយើងដឹងតិចតាមរយៈការបង្ហាញ និង Maximum Voltage វានឹងបង្ហាញបានប្រើប្រាស់កុងដី។
- **Leakage Current** : កុងដីមិនមែនល្អឥតខ្ចោះទេ វារាយមានចរន្តមួយចំនួនត្រូវបានប្រភេទ កាត់ខិះអេឡិចត្រូច ។ ចរន្តនេះគេហោថា Leakage ។ Leakage រាយបង្ហាញ អស់ចាមពល បណ្តុះរងគុងកុងដី ។
- **Equivalence Series Resistance (ESR)** : ចង្វាប់ទាំងពីរនៃកុងដីមិនមែនចំលង អតិសនិ 100% ទេ វារាយមាននសិស្សដែលគួរបាន (ប្រែហល 0.01 Ω) គឺរាយបង្ហាញនៅ ពេលមានចរន្តដំឡូងកាត់ហើយបានក្នុងការបង្ហាញ និងខាតរាយការណ៍ ។
- **Tolerance** : គឺជាកម្រិតលេរ្ព័ន់នៃកុងដីខុសត្រាគ់តាមប្រភេទនីមួយា ។ ជាទុទៅមាន ចន្លោះចាប់ពី $\pm 1\%$ ទៅ $\pm 20\%$ ។

២.២. រួមទៅកាន់កុងដីទៅប្រភេទខ្ពស់ប្រចាំឆ្នាំ

- **Electrolytic Capacitor (កុងដីមានបូល)**



រូបភាព ១០ : របៀបម៉ែនដើងកុងដីប្រកាន់បូល (អេឡិចត្រូលិត)

- កុងដីប្រកាន់នេះរាយជាប្រកាន់ស្ថាមផ្ទុកខ្ពស់ប្រចាំឆ្នាំ ដែលសារតាមរយៈការបង្ហាញ ។
- តំលៃ Capacitance ដែលបានបង្ហាញឡើង និងប្រកាន់បូល ។ ដូចនេះយើងត្រូវប្រើប្រាស់ការបង្ហាញប្រចាំឆ្នាំ ។

- Maximum Voltage និង Capacitance ត្រូវបានបញ្ជាផ្ទៃក្នុងបច្ចុប្បន្នបញ្ជាផ្ទៃក្នុងចំណាំនៃការបង្ហាញ Datasheet ។

➤ កុងដង់សេវាគិច្ច (Ceramic Capacitor)



រូបភាព ១១ ៖ កុងដង់សេវាគិច្ច (Ceramic Capacitor)

- កុងដង់សេវាគិច្ចជាកុងដង់តំលៃបេរិចនប្រកាន់បូលហើយប្រើបានកុងដង់សេវាគិច្ចជាទីអនឡិចត្រួច
- កុងដង់ប្រភេទនេះមាន Capacitance តួច ៧ និងត្រូវរាយលេខកុងដង់ដឹងតំលៃរបស់វា
- ការអានតម្លៃកុងដង់សេវាគិច្ច



រូបភាព ១២ ៖ របៀបអានតម្លៃ កុងដង់សេវាគិច្ច

ឧបាទាង័ែ៖

➤ កុងដង់សេវាគិច្ចដែលបង្ហាញតម្លៃ 474



រូបភាព ១៣ ៖ របៀបសេវាគិច្ចមានបង្ហាញលេខ

យើងបាន : Capacitance : $47 \times 10^4 \text{ pF}$
 $= 470000 \text{ pF} = 470 \text{ nF} = 0.47 \mu\text{F}$

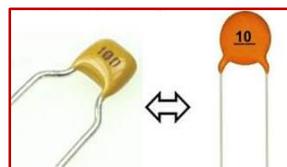
- កុងដង់សេវកិមិចដែលបង្ហាញតម្លៃ 102K 2KV



យើងបាន : Capacitance : $10 \times 10^2 \text{ pF} = 1000 \text{ pF} = 1 \text{ nF} = 0.001 \mu\text{F}$,
Tolerance : K = $\pm 10\%$ និង Maximum Voltage 2KV = 2000V

- កុងដង់សេវកិមិចដែលបង្ហាញតម្លៃ 100

យើងបាន : Capacitance : $10 \times 10^0 \text{ pF} = 10 \text{ pF}$



ភាគច្រើនកុងដង់សេវកិមិចត្រូវបាន 100pF យើងអាចដឹងពី Capacitance ដោយធ្វាប់បាន។



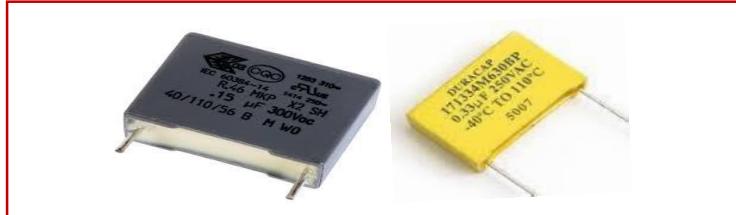
- កុងដង់មីក



រូបថាន ១៤ : រូបពិតកុងដង់មីក

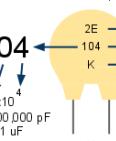
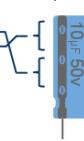
- កុងដង់មិកាតាកុងដង់តាំលេចខ្លួន មិនប្រកាន់បូល
 - កុងដង់ប្រកែទនេះ Maximum Voltage តាំលេ គីអាប្រើយទៅលើផ្ទុរបស់ភានិងត្រូវអានលេខកុង តាំលេរបស់ភាពុចកុងដង់សេវាកម្មិចដែល។
 - ចំពោះផ្ទុ ពណ៌ក្រុហម 250V ពណ៌ត្រូវត 100V ពណ៌លើឃ 400V ពណ៌ទីក្រុច 300V ពណ៌លើបែកង 500 V ពណ៌ខ្សែវ 600V ។

➤ កុងដែងហើលឆ្នាសីច



គ្រោង ១៤ ៖ រូបពិតកុងដៃ ហើលឆ្វាស្ទិច

- តើយកប្រភេទហើលប្លាសិកមួយចំនួនទៅធ្វើជាតីអេឡិចត្រូវបស់វា ។
 - កុងដងប្រភេទនេះអាចជាប្រភេទសូតិង ជាប្រភេទសែម បុមានសារធាតុរវាង ។
 - Maximum Voltage និង Capacitance ត្រូវបានពេញតម្លៃត្រូវបានបង្ហាញនៅលើក្នុង ប្រចាំថ្ងៃនៃការបង្ហាញ Datasheet ។
 - គឺជាប្រភេទមិនប្រកាសប៉ូល ។

Capacitors		
Ceramic Capacitor	Electrolytic Capacitor	
 <p>104 10×10^4 $= 100,000 \text{ pF}$ $= 0.1 \mu\text{F}$</p> <p>Symbol (Non-Polarized)</p>	 <p>2E 104 K Max. Voltage Capacitance Tolerance</p> <p>1010J50V Symbol (Polarized)</p>	
Capacitance Conversion Values		
Microfarads (μF)	Nanofarads (nF)	Picofarads (pF)
0.000001 μF	0.001 nF	1 pF
0.00001 μF	0.01 nF	10 pF
0.0001 μF	0.1 nF	100 pF
0.001 μF	1 nF	1,000 pF
0.01 μF	10 nF	10,000 pF
0.1 μF	100 nF	100,000 pF
1 μF	1,000 nF	1,000,000 pF
10 μF	10,000 nF	10,000,000 pF
100 μF	100,000 nF	100,000,000 pF
Max. Operating Voltage		
Code	Max. Voltage	
1H	50V	
2A	100V	
2T	150V	
2D	200V	
2E	250V	
2G	400V	
2J	630V	
Tolerance		
Code	Percentage	
B	$\pm 0.1 \text{ pF}$	
C	$\pm 0.25 \text{ pF}$	
D	$\pm 0.5 \text{ pF}$	
F	$\pm 1\%$	
G	$\pm 2\%$	
H	$\pm 3\%$	
J	$\pm 5\%$	
K	$\pm 10\%$	
M	$\pm 20\%$	
Z	+80%, -20%	

ផ្សេងៗ ១៦៖ របៀបអានកម្លា កុងដីដែលបានចាប់ផ្តើមនៅក្នុងប្រព័ន្ធផ្សព្វផ្សាយ

➤ គុណភាព នៃថ្វីប្រព្រល

កុងដីផែតម៉ែប្រព្រលគឺជាកុងដីដែលយើងអាចធ្លាស់បួរកំលែល Capacitance របស់វាបាន ដោយ ធ្លាល់បាន ។ កុងដីនេះចំការដាន គឺ Tuning Capacitor និង Trimmer Capacitor គឺត្រឹមត្រូវ ក្នុងវិទ្យា ដើម្បីបួរប្រកង់ប៉ុស្តី ។



រូបភាព ១៧ ៖ កុងដីផែតម៉ែប្រព្រល

៣. ឌីອីដ (Diode)

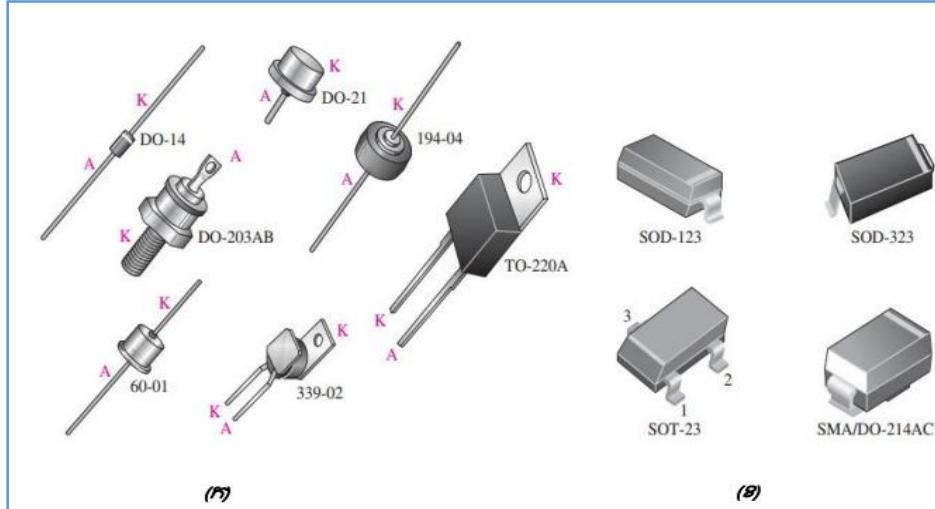
ខីយូតការមានសមត្ថភាពចំលងចរន្តកុងទិសដោមួយគត់និងនៅខាងក្រោម: ដែលវាមិនអនុញ្ញាតឲ្យចរន្តរត់ កាត់កុងទិសដោរឃ្លានេះទេ ដែលវាមានអត្ថប្រយោជន៍សម្រាប់ដំណើរកុងប្រភេទស៊ីតី ឬឡើចត្រូនិច ឱ្យបានប្រើប្រាស់ ។ ខីយូតក្រុវាបានគេយកទៅប្រើប្រាស់ធ្វើជា AC rectifier, diode Limiter, diode clampers និងdiode voltage multipliers ។

៣.១. ចំណើនភាពសំខីយូត

ខីយូតគឺជាគ្រៀងពាក់កណ្តាលចំលងដែលមានគោលពីរត្រូវបានបង្កើតឡើងដោយ តំបន់ doped ចំនួនពីរដែលរួចរាល់ចំណុចជាប់ចេញពីគ្នាដោយ PN junction មួយ ។ នៅចំនួននេះ យើងនឹងសិក្សាទី របៀបប្រើប្រាស់ការស្វែងរកមួយខ្លួន ឱ្យបានបង្កើតឡើង ។ ដំណើរការនេះត្រូវបានគេ ហោចា ការធ្វើបូលកម្ម (biasing) ។ ការធ្វើបូលកម្មរបស់ខីយូតមានពីរគឺការធ្វើបូលកម្មស្រប(Forward Bias)និងប្រាស់ (Reverse Bias) ។

៣.២. ប្រភេទខ្សែសង្គម

នៅក្នុងរូបភាព ១៤ (ក) អាជីវការ និងកាតុត គឺត្រូវបានបង្ហាញនៅលើ ខីយីតតាម លក្ខណៈ ដើម្បីបង្កើតការស្រែយោទនៃប្រភេទនេះ ខីយីតនឹងមួយយ៉ា កាតុតជាទុកដាក់ត្រូវបានគេសំគាល់ដោយមានតំនុសង្គត ប្រហែលនៅលើពីលីតិលីតិលីកលេងតែប្រភេទ DO-203AB ។



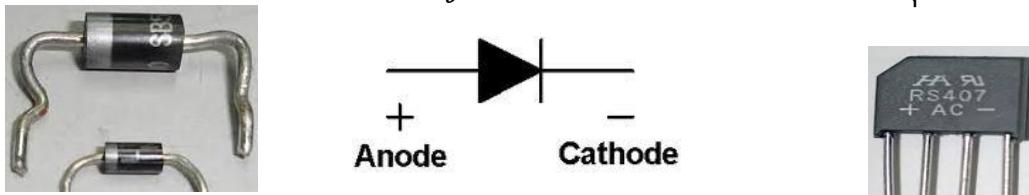
រូបភាព ១៥ : ប្រភេទខីយីត

នៅក្នុងរូបភាព ១៥ (ខ) បង្ហាញពីប្រភេទនៃខីយីតសម្រាប់ Surface Mounting នៅលើ Printed circuit board (PCB) ។ SOD និង SOT package មានរាងដូចជាដើរីសត្វ ហើយ SMA package វិញ្ញាមានរាងជាមក្ស L ។ SOT និង SMA ជាប្រភេទដែលមានតំនុសង្គត សំគាល់នៅផ្ទុកខាងចុងដែលជាដើរីសត្វ ។ SOT មានដើរីបី ដែលនៅក្នុងនោះមានខីយីតមួយ បុពុរ ។

៣.៣. ប្រភេទខីយីតដែលប្រើប្រាស់នៅក្នុងការបង្កើតប្រព័ន្ធ

៣. ខីយីតដែលប្រើប្រាស់នៅក្នុងការបង្កើតប្រព័ន្ធ

គឺជាផីត្រាប្រាជ់មានតូនាទីសំរាប់ប្រាជ់បរន្ទូអតិសនីពីចរន្ទACទៅDC, ACទៅAC, DCទៅAC និងDCទៅDC ។ ខីយីតប្រភេទ យីតដែលត្រូវបានគេសិក្សាប្រើប្រាស់ដូចខាងក្រោម និង Power Supply



រូបភាព ១៦ : ខីយីតបំលែងក្នុងប្រព័ន្ធប្រើប្រាស់

២. ឌីយ៉ូត Zener

ឌីយ៉ូត Zener មានត្បាទីសម្រាប់ធ្វើឲ្យចរន្តមានល្អរភាពឡើង និងមានត្បាទីសម្រាប់ធ្វើឲ្យមានកុលយោរាល និងសិកុណ្ហភាព ដែលធ្វើអោយសំលេងខ្ពស់ទាប បំបាត់សំលេង រខាន នៃប្រភេទ និងមានត្បាទីធ្វើឲ្យរបភាពនៃបណ្តាញទូទៅស្ថិតនឹងនរបំបាត់សំលេងធ្វើដែលផ្តល់នូវការតាមចរន្តអគ្គិសនី។

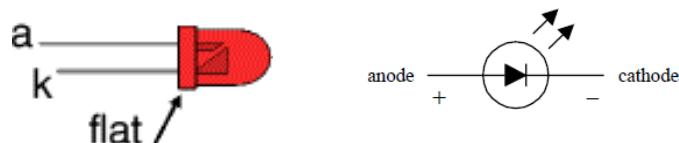


រូបភាព ២០ ៖ ឌីយ៉ូត Zener ប្រហែលយើតកែវ និងនិមិត្តសញ្ញា

៣. ឌីយ៉ូត ពន្លឹ (Light-Emitting Diodes(LEDs))

ឌីយ៉ូតនឹងប្រហែល LED គឺជាប្រភេទបិទ្ទាមួយដែលគេនិយមប្រើយ៉ាងទូលំទូលាយ។ light-Emitting Diodes(LEDs) គឺមានពន្លឹសាយចេញពី PN junction នៅពេលមានចរន្តផ្តើមការតែខែដោយ ពេលចូលកម្មស្រប។

LEDs ប្រភេទនេះ គឺវាទិនមានតម្លៃថ្មីទេ ហើយវាកើតឡើងដែលធ្វើឲ្យដែលធន។ ឌីយ៉ូតប្រភេទនេះ គឺគេបានប្រើ សម្រាប់បញ្ជាញពន្លឹ។

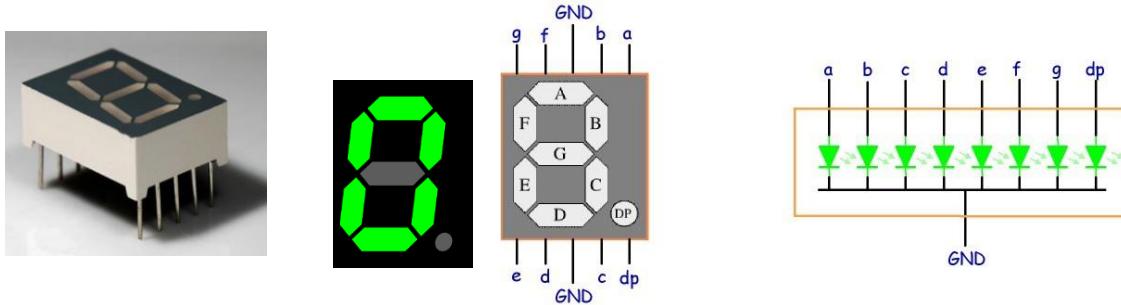


រូបភាព ២១ ៖ ឌីយ៉ូត ពន្លឹ ប្រហែល LED និងនិមិត្តសញ្ញា

៤. 7-segment Display

7-segment Display គឺត្រូវបានប្រើដើម្បីបំភ្លើលេខនិងអក្សរហើយត្រូវបានប្រទេសរួចរាល់ និងនាងិក ឱ្យធ្វើចិត្ត ម៉ាសីនគិតលេខខ្សោច ម៉ាសីនរៀង VCD និងក្នុង ធមិតិផលអេឡិច្ច

ត្រូនិចជាប្រើប្រាស់ 7-segment Display គឺកញ្ចែងពី LED ចំនួនប្រាំបីគ្រាប់ តាំងរៀងលេខ "8" ។ ចំនួនលេខ 0 - 9 A B C D E F អាចត្រូវបានបំភ្លើ ដោយការផ្តាស់ប្តូរឡើ Led ។ ការក្រុមរបស់បណ្តាល 7-segment Display នេះ ហេចចា ការក្រុម ។ ដូចត្រូវដែរចំពោះប្រកែទ អាណាព្យម គឺដើរអាណាព្យតាំងអស់ត្រូវបានជាប់ត្រា។



រូបភាព ២២ ៖ 7-segment និងក្នុងនាទីដើរនឹងនីមួយៗ

៤. ត្រួតតិច (Transistor)

ត្រួតតិចស្ថិស្សន៍ គឺជាប្រព័ន្ធឌែលបានគេយកមកប្រើបានការប្រើប្រាស់បច្ចេកទេស និងស្រួលបាន ពង្រីកអាជីវការ និងជាកុងតាក់អេឡិចត្រូនិច ។ ត្រួតតិចស្ថិស្សន៍រួចរាល់ជាប្រភេទ គឺ N-P-N និង P-N-P ។



រូបភាព ២៣ ៖ ប្រភេទត្រួតតិចស្ថិស្សន៍

❖ គារប្រើប្រាស់តិចិត្តស្ថិតិស្ថិតិ

នៅត្រប់ត្រូវបស់គ្រឹងអេឡិចត្រូនិចគឺតិចិត្តស្ថិតិដែលត្រូវបានតាំងលើលេខសំគាល់រីពណ៌ដើម្បីសំគាល់ ពីប្រភេទ របស់ គ្រឹងនិមួយៗដោយទេរូកត្រប់ត្រូវគឺតិចិត្តស្ថិតិគឺត្រូវបានលើខសំគាល់ថា ជាប្រភេទPNP ឬ NPN គឺយក តាំងលើកូដនោះទៅ Search in Google រកប្រភេទ Datasheet របស់វានោះយើងដឹងថា វាតារគ្រឹងប្រភេទណី។

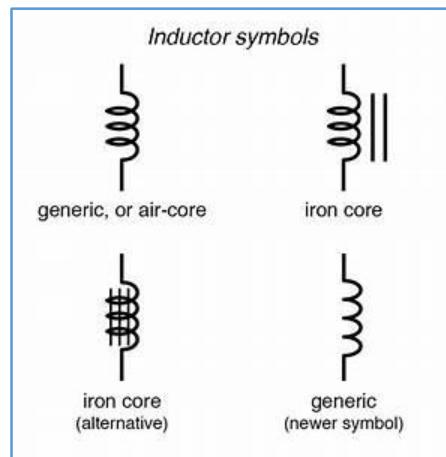
៥. អំឡុងតុល់ (Inductor)

វាតិជារបីនៃខ្សែចំលួងមានអីសីទ្វាន់ស្ថិតិរុបាស្ទីលើអង្គធាតុដែលអាចមានជម្រូបម៉ាញ្ញីទិចបាន មានផ្ទុចជានេះ លោកេ: ដែក , Ferrite។លើ។ នៅពេលដែលមានលំហ៊ូចនូននៃ កាត់របីការងារ អុចតង់បង្កើតដែនម៉ាញ្ញីទិច ហើយដែនម៉ាញ្ញីទិចនេះមិនងាយនឹងឆ្លាត់ស្ថិតិ លក្ខណៈ របស់វានេះ ជាលទ្ធផលនៃលក្ខណៈពិសេសនេះការងារអុចតង់ផ្ទុកចាមពលក្នុងរបីរបស់វា ។ ឧទាហរណ៍នៅពេលដែលចរន្តនៃកាត់របីការងារអុចតង់ប្រើប្រាស់ប្រើប្រាស់គឺតិចិត្តស្ថិតិមួយរក្សាថ្មីនៅមេរោគ។

ផ្ទុចនេះ ការងារអុចតង់ប្រើប្រាស់ម្រាប់ស្ថិតិរាតចននេះ ។



រូបភាព ២៤ ៖ ប្រភេទការងារអុចតង់



រូបភាព ២៥ ៖ សិមិត្តសញ្ញាប្រភេទការងារអុចតង់

ផែង្វែលទី២ ៖ ឧត្តមេ (Battery)



រូបភាព ១៖ រូបពិតនៃប្រភេទបាត់គេរីជួយ។

៩. ផ្ទើផែវបាត់ ឧត្តមេ

បាត់គេរីជាប្រភេទធ្វើតែតុងតង់ស្ថិតិថាប៉ាប់សម្រាប់ខែករណ៍អេឡិចត្រូនិច ហើយជាលំយានយន្តនិងស្ថានីយ៍ផ្សាយទូទៅមនាគមន៍ ។ បាត់គេរីដែរឯងជោយស៊ិលធូកតង់ស្ថិតិ (voltaic cells) បុស៊ិលអេឡិចត្រូតីមី (electrochemical cell) មួយបុរីន៍ ។ កម្រិតតង់ស្ថិតិចេញ និងអាជ្ញាន់ចរន្តរបស់បាត់គេរីក្រែសំយនិងកត្តាប្រើប្រាស់ វគ្គបាត់ដើមនៃអេឡិចត្រូត រួចរាល់ និងទំហំនៃអេឡិចត្រូត ។

បាត់គេរីទូទៅត្រូវបានបែងចែកជាតី គឺ Primary cells ឬ Disposal Battery និង Secondary cell ។ Primary cells ជាប្រភេទស៊ិលដែលមិនអាចផ្តល់បន្ទូកចូលវិញ (recharge) ពេលប្រើប្រាស់អស់ មាននំយច្ចាមិនអាចបែងចែកជាតីឡើង ។



រូបភាព ១ ៖ Primary Cell ប្រភេទស្ថិតិ

Secondary Cells អាចធ្វើលើបន្ទុកឡើងវិញបាន ពីព្រះប្រព័ន្ធឌីជីមានលក្ខណៈបច្ចាស់ ។ នៅពេលធ្វើបន្ទុកឡើងទូលាមាននៃយចាត់សិលបំលែងចាមពលគឺមិនត្រូវឡើងដោយពលអត្ថិសនី ពេលទូលាបន្ទុក (សាក) មាននៃយចាត់បំបែងពីចាមពលអត្ថិសនីឡើងដោយពលគឺមិនធម្មិញ ។



រូបភាព ២ : Secondary Cell ឬ Rechargeable Battery

នៅពេលសាកសែលដើរត្រូវជាសុវត្ថិភាពបន្ទុក ដែលត្រូវបានធ្វើឡើងដោយប្រភពធ្លើត្រូវឡើង ជាប់ ពីខាងក្រោម ។ ការធ្វើវានឹងការទូលាបន្ទុកឡើងវិញបែបនៃសែលគេហែង Cycling ។ គេភាពហេរ Secondary Cell ឬនានមានឯកសារឡើងឡើងតិច Storage Cell ។

តារាង ១ : បាត់ត្រូវរួមចំណែក និងកម្រិតកង់ស្ថាយ

ឈ្មោះសែល	ប្រភេទ	កង់ស្ថាយពេលត្រូវបន្ទុក (V)
លីចូម	Primary	3
កាបូន-ស៊ីដុសី	Primary	1.5
ស៊ីដុសីក្រឹត	Primary	1.5
មង់កាណាលីអុកសុីត (អាលកាញរំង)	Primary/ Secondary	1.5
បារិតអុកសុីត	Primary	1.35
ប្រាក់អុកសុីត	Primary	1.5
អីយ៉ូងលីចូម	Secondary	3.7
សំណាក់រាសុីត	Secondary	2.1
នីកែល-កាង្យម	Secondary	1.2
Nickel-Metal-Hydride	Secondary	1.2
សែលនីកែល-ដែក (Edison)	Secondary	1.2
នីកែល-ស៊ីដុសី	Secondary	1.6
សុទ្ធនា	Secondary	0.5

កុងដឹងទំហំ (Supper Capacitor) អាចប្រើបានសាកត់តែវិញ្ញានក្នុងករណីខ្លះ ប៉ុន្មោះ ប៉ុន្មោះគេពិតាកដលិតនិងមានតម្លៃថ្មី។ កុងដឹងសាកល្បៈនឹងធានាថាដែរពីរបាយ៖វាមិនមានប្រព័ន្ធគ្មតិមីទេ ប៉ុន្មោះកុងដឹង ផ្តល់បន្ទុកអស់លើឯងជានាថែរ។



រូបភាព ៣ : Supper Capacitor

២. សេវានឹងថ្មី

សេវានឹងថ្មីមានព័ត៌មានស្ថិតិយវត្ថុ មានអាយុកាលខ្ពស់ និងទំហំក្នុង សេវានឹងថ្មីនេះអាចដាក់ចាមពលប្រើបានដោយសេវានឹងថ្មីសំខាន់សំខាន់បន្ថែមទាំងអស់ ដែលបានប្រើបានក្នុងការប្រើប្រាស់ក្នុងរយៈពេលរហូតដល់ ១០ឆ្នាំ និងបានប្រើបានដូចជាលិមិនីអូកសុំត (LiSO₂) និងបីចុមក្នុវិត ។



រូបភាព ៤ : ប្រើប្រាស់បីចុម

៣. អាតុយ (ស៊ីនសំខាន់-អាតុយ)

អាតុយត្រូវបានចាត់ទុកជាតា Secondary Cells ។ ពីព្រោះវាអាចសាកបន្ទុកឡើងវិញបាន ។ អាតុយអាចផ្តល់បន្ទុកដំឡើង ។ ឡើងត្រូវឱ្យការប្រើប្រាស់ឡើងដោយសូលូយសូលូវកាសីតករជាតិសែស អាសីតសិុលផ្លូវិច (H₂SO₄) ។ គេនិយមប្រើប្រាស់ក្នុងការបញ្ចុះរចយន ចរន្តដែលត្រូវការដើម្បី ចាប់ធ្វើម ពីចន្ទោះ 50 ទៅ 400A ។ ជាទុកអាតុយមួយអាចធ្វើឡើងដោយប្រើប្រាស់សែល ក្នុងមួយ សែលមាន តង់ស្រួលណូមិោល 2.1V ។ ដូចនេះដើម្បីបានអាតុយដែលមានតង់ស្រួល 12 V គគ្រោរការ 6 សែល តាមសេវីនីក្តា ។

ចំពោះអាតុយយានយន្តយើងអាចធ្វើរបន្ទុក និងសាកបន្ទុកបានជាប្រើប្រាស់ដឹងក្នុងរយៈពេល ៣ ទៅ ៥ ឆ្នាំ ប្រសិនបើឱលក្នុណ៍: សិក្សាពាណិជ្ជកម្មរក្សាទុក ។



រូបរាង ៥: អាតុយ 12V ប្រើប្រាស់ក្នុងយានយន្ត

៣.១ អ្នកចារន៍

អ្នកចារន៍គឺជាលំហូរនៃចរន្តក្នុងកំឡុងពេលជាក់លាក់មួយ គេកំណត់ថា អំពេ-ម៉ោង (A.h)។ ឧបាទរណី អាតុយចរន្តដែលមានអ្នកចារន៍ 200A.h មាននំយចាប្រសិនបើយើងធ្វើរបន្ទុកកម្រិត 25A នោះ វានឹងធ្វើរក្នុងរយៈពេល 200/25 គី 8ម៉ោង ។ បើនេះកាលណាយើងធ្វើចរន្តកាន់តែជំនោះរយៈពេល នៃការធ្វើអេស គឺកាន់តែបាប់ ។ ជាទុកទៅគេកំណត់ឲ្យប្រើអាតុយក្នុងសិក្សាពាណិជ្ជកម្មពី25ទៅ26 អង្វាស់។ សិក្សាពាណិជ្ជកម្មកាន់តែខ្ពស់កាន់តែជម្រើសលេរ្យនៃប្រពិកម្ម បើនេះសិក្សាពាណិជ្ជកម្មកើនលើស 45°C អាចឲ្យ អាតុយបាប់ខ្ពចំ។ បើសិនសិក្សាពាណិជ្ជកម្មចុះបាបនាំម្រោមនាការបន្ថែមយចន៍ និងតង់ស្រួល ។ នៅសិក្សាពាណិជ្ជកម្ម ១៥ °C អ្នកអំពេម៉ោងធ្លាក់ចុះ 40 % ។ ដូចនេះមុននឹងយកទៅប្រើក្នុងកន្លែងដែលមានសិក្សាពាណិជ្ជកម្ម យើងត្រូវសាកុំវាបានពេញមុនសិន ។

៣.២ ឧត្ថមីផែនគ្មោះ

ដង់សុំតេបញ្ញាកំពើលក្ខណៈដោរនៃភាពឃុំ ។ ដង់សុំតេជាជាលធ្វើបន្ថែមភាពឃុំនិងទីក ។ ដង់សុំតេនៃភាសុំតុលាលួរិចជាងដង់សុំតេទីក 1.85 ដង់ ។ គេចាត់ទុកដង់សុំតេទីកតីស្តី 1 ។

ចំពោះភាពឃុំដែលសាកពេញរាមានដង់សុំតេ 1.280 នៅសិតុណ្ឌភាពបន្ទូប់ (21 ទៅ 25 អង្វាស់) នៅពេលដែលសិលដ្ឋូរបន្ទូកនោះ ទីកមានការកើតឡើងត្រឹមត្រូវ ដូចនេះដង់សុំតេញាកំចុះ នៅពេលដែលដង់សុំតេក្នុងជាង 1.145 គេចាត់ទុកចាត់សិលបន្ទូរអស់ ។ ដើម្បីរារស់ដង់សុំតេត្រប្រើ Hydrometer ហើយដង់សុំតេរបស់ភាពឃុំសំណាកសុំតេ តី : $V = \text{ដង់សុំតេ} + 0.84$ ។

ឧទាហរណ៍ ចំពោះដង់សុំតេ 1.280 នោះ $V = 1.280 + 0.84 = 2.12V$ ។ តែម្រោងនេះតង់ស្រួលភាពឃុំពេញបន្ទូក ។



រូបភាព ៦៖ Hydrometer ប្រើសម្រាប់រារស់ដង់សុំតេភាពឃុំសំណាកសុំតេ

៤. សុឡានថែងជែង

សុឡានសិល (Solar cell) គឺជាការបំលែងចាមពលពន្លឹះព្រះភាពឃុំ ឲ្យទោជាចាមពលអគ្គិសនី។ សិលរបស់វា ផ្តើមជាយីដិជ្ជីកុងខុចទៅ ដែលរាបដ្ឋីតាមដង់សុំតេដែលបានបង្កើតឡើងចេញនៅពេលបន្ទូលពន្លឹះ។ សុឡិសូមជាតាតុដែលត្រូវគេកំពុងប្រើប្រាស់ការប្រើប្រាស់ការបង្កើតដង់សុំតេដែលបានបង្កើតឡើងចេញប្រមាណ 0.5V ។ ចំពោះការប្រើប្រាស់គេងសិលជាប្រើប្រាស់នៅក្នុង module ទៅមួយដើម្បីអនុវត្តបង្កើតបានតាមតម្លៃការ។

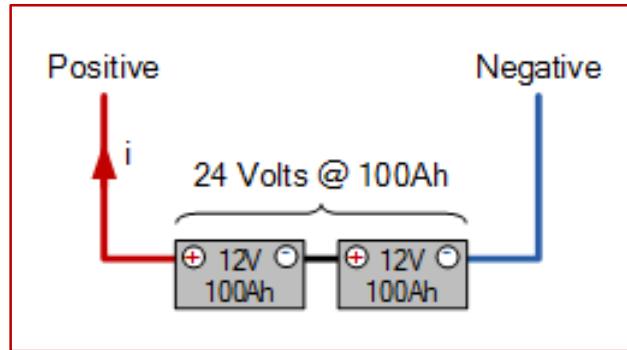


រូបភាស ៧៖ សុឡានសិលប្រកែទេ Crystalline silicon

៥. ការតាត់តែវិជ្ជាស៊ី និងខ្វោច

៥.១ ការតាត់តែវិជ្ជាស៊ី

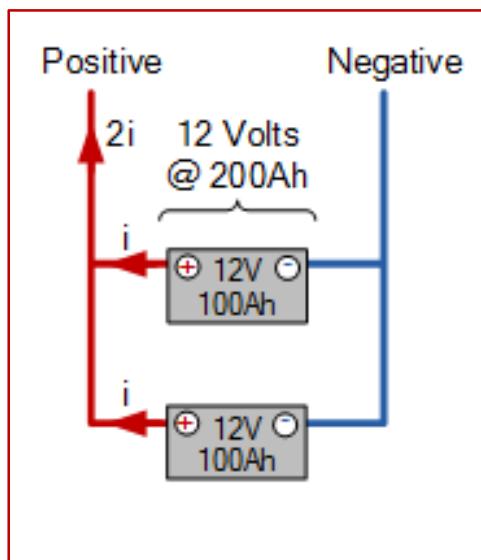
ការតាត់តែវិជ្ជាស៊ីបង្កើនតង់ស្បែងចេញ ។ តង់ស្បែងចេញសរបតីស្ទើសុំនឹងផលបូកតង់ស្បែង របស់បាត់តែវិជ្ជាស៊ីមួយ។ វិនិច្ឆន៍របស់វារឿងតីស្ទើសុំនឹងចរន្តនៃបាត់តែវិជ្ជាស៊ីមួយនៅ៖ ។



រូបភាព ៥៖ ការតាត់តែវិជ្ជាស៊ី

៥.២ ការតាត់តែខ្វោច

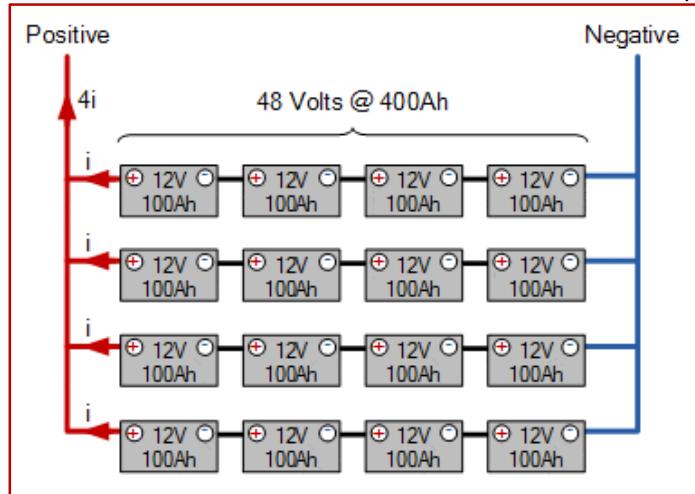
ការតាត់តែខ្វោចគឺបង្កើនចរន្ត ។ ចរន្តសរបតីស្ទើសុំនឹងផលបូកចរន្ត របស់បាត់តែវិជ្ជាស៊ីមួយ វិនិច្ឆន៍របស់វារឿងតីស្ទើសុំនឹងតង់ស្បែងនៃបាត់តែវិជ្ជាស៊ីមួយនៅ៖ ។ ជាទាមរណ៍ យើងតាត់តែវិជ្ជាស៊ី 12V ដែលផ្តល់ចរន្ត 100Ah ម៉ោងនៃពីរ នៅវានឹងផ្តល់ចរន្ត 200Ah និងតង់ស្បែង 12V ដែល ។



រូបភាព ៦៖ ការតាត់តែខ្វោច

៥.៣ ការសរស់ស្ថិតិខ្លួន

របាយការក្រោមគឺជាការតិចតាត់តែវិជ្ជាស៊ីនិងខ្សោង ដើម្បីបង្កើនការស្វែងរកនិងចរណី



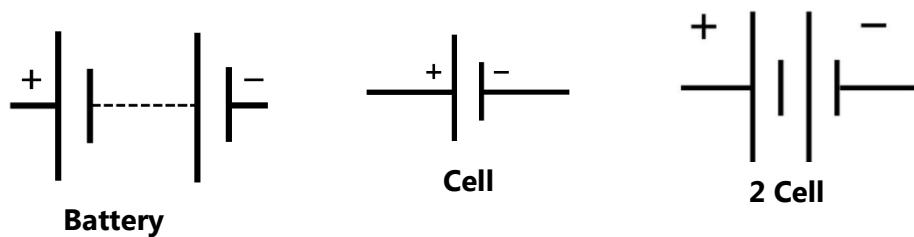
រូបភាព ៩០ ការតិចតាត់តែវិជ្ជាស៊ីនិងខ្សោង

៦. តារាងប្រភេទបញ្ហាកម្មណ៍

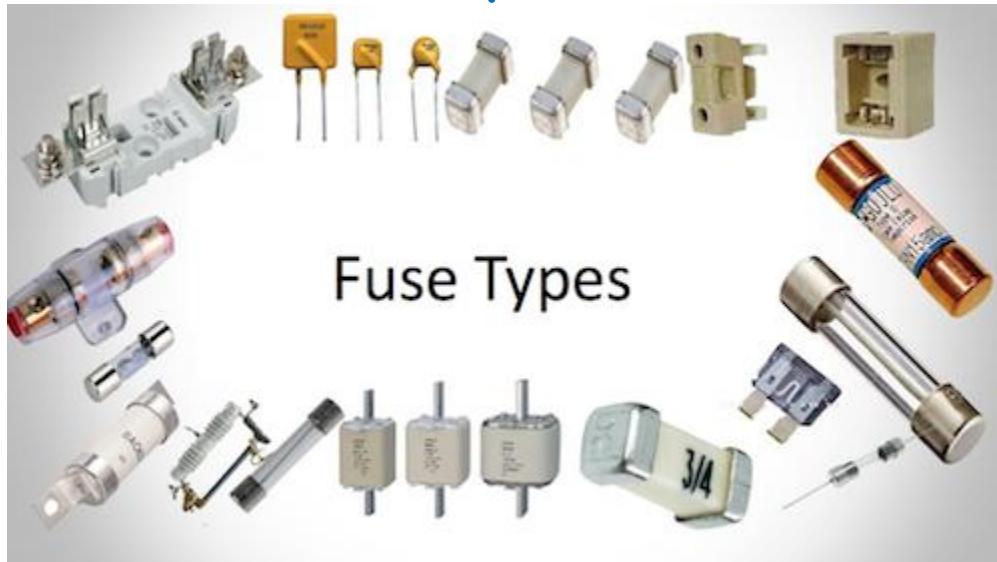
Type	Typical Capacity	Voltage	Max. Discharge Current	Features	Common Uses
Lithium button cell (e.g., CR2032)	200mAh	3V	4mA with pulses up to 12mA	High temperature range (-30 to 80°C); small	Low-power devices; RF remote controls; LED key ring lights; etc.
Alkaline PP3 battery	500mAh	9V	800mA	Low cost; readily available	Small portable electronic devices; smoke alarms; guitar pedals
Lithium PP3	1200mAh	9V	400mA pulses up to 800mA	Expensive; light; high-capacity	Radio receivers
AAA cell	800mAh	1.5V	1.5A continuous	Low-cost; readily available	Small motorized toys; remote controls
AA cell	3000mAh	1.5V	2A continuous	Low cost; readily available	Motorized toys
C cell	6000mAh	1.5V	Probably get away with 4A	High-capacity	Motorized toys; high-powered flashlights
D cell	15,000mAh	1.5V	Probably get away with 6A	High-capacity	Motorized toys; high-powered flashlights

Type	Typical Capacity	Voltage	Features	Common Uses
NiMH button cell pack		80mAh	2.4 or 3.6V	Small Battery backup
NiMH AAA cell		750mAh	1.25V	Low cost Replacement for single-use AAA cell
NiMH AA cell		2000mAh	1.25V	Low cost Replacement for single-use AA cell
NiMH C cell		4000mAh	1.25V	High capacity Replacement for single-use C cell
Small LiPo cell		50mAh	3.7V Low cost; high capacity for weight and size	Micro-helicopters
LC18650 LiPo cell		2200mAh	3.7V Low cost; high capacity for weight and size; slightly bigger than AA	High-power flashlights; Tesla Roadster (yes, really—about 6800 of them)
LiPo pack		900mAh	7.4V Low cost; high capacity for weight and size	Cell phones, iPods, etc.
Sealed lead-acid battery		1200mAh	6/12V Easy to charge and use; heavy	Intruder alarms; small electric vehicles/ wheelchairs

៤. ប្រព័ន្ធលិថែនសញ្ញា



ເສດຖະກິດ ຂະໜາ : ສູງແຫຼມ (FUSE)

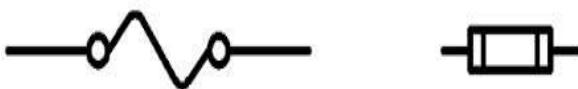


រូបភាព ១ ៖ រូបពិតនៃប្រភេទក្នុយសុបធ្លឹង

୧. କ୍ଷେତ୍ରପାଳ୍ଯମାନ

ក្នុងសៀវភៅអតិថិជន បុសៀវភៅអនុញ្ញាតត្រីមួយសីប គឺជាបន្ទូលមានពិន្ទានទីសម្រាប់ការពារគ្រឹងទូទៅប្រភពពីការពារសៀវភៅមិនច្បាស់លើក្រុមហ៊ុនប្រើប្រាស់លើសលប់(overload)នៅពេលមានលើក្រុមហ៊ុនលើសអត្រាបន្ទូលបានកំណត់របស់ក្នុងសីបខ្លួចម្អោងខាងក្នុងនិងរលាយដាច់ហើយដាច់សៀវភៅទាំងស្រុង។ សៀវភៅផ្តើមបន្ទូកសីអាណកាតជំការពាក្យាប់មិនបានត្រីមក្រវនិងដំណើរការ របស់ខ្លួនជាមួលហេតុដែលបណ្តាលច្បាស់លើក្រុមហ៊ុនប្រើប្រាស់លើសលប់។

ភ្នែកយសុធម៌បានជាមួយ សៀវភៅចរន្តជាប់ និងចរន្តនាស់ ។ ភ្នែកយសុធម៌បានប្រើប្រាស់ តាមតម្លៃការជាទុទិនិជ្ជាថ្នាប់ខ្សោយព្យាប់ នៅពេលដែលមានលំហ្អចរន្តលើសពីអត្រាដែលបានកំណត់ក្នុងរយៈពេលខ្លឹមបំផុត ។



រូបភាព ២៖ និមិត្តសញ្ញាណនៃភ្លាយសុប្បរិយា

នៅឆ្នាំ 1890 លោក Thomas Edison បានចាត់ទុកថាគ្មួយសីបជាដៃដើរកម្មយនៃប្រព័ន្ធដែកចាយអគ្គិសនី របស់គាត់ ហើយគាត់បានកាន់កាប់សិទ្ធិថាតីផង ។

២. សំណាគត់លេខយសីប

ចុងគត្រាប់ទាំងពីរធ្វើពីលោហេ: ដែលមានមុខភាពថា ខ្លួន ឬ ខ្សោយសីប ដែលត្រាប់រាយការចុងទាំងពីរធ្វើពីលោហេ: ត្រូវបានប្រកែទមានដូចជា : ស៊ីនីសី ឡើងដែង ប្រាក់ កាលុយមិញ្ញម... ឬ ខ្សោយសីប ដែលមានលំហូរចរន្តលើសលប់ ឬ ខ្សោយសីប ដែលមានសុវត្ថិភាព ហើយវានឹងបានចុងក្រោយការប្រាក់ដែលមានចរន្តផ្ទាល់ខ្លួនកាត់ ឬ បរិមាណនៃកំដៅដែលកាយអារ៉ាស្រីយនឹងទំហំនៃក្សួយសីប ឬ ក្សួយសីប ដែលមានចរន្តផ្ទាល់ខ្លួនបំពុំដើរ បុរាណកំដើរដើរការការកាត់ដាច់ ឬ



រូបភាព ៣៖ សំណាគត់នៃក្សួយសីប

៣. ប៉ាវតែងត្រួមយកដំឡូលលេខយសីប

៣.១. សញ្ញាផ័ត៌មាន

នៅលើចុងគត្រាប់ទាំងពីរនៃក្សួយសីបភាពត្រូវបានគេដាក់សញ្ញាសម្រាប់អត្រាចរន្តនិងតង់ស្រួល ។



រូបភាព ៤៖ ក្សួយសីបមានសញ្ញាសំគាល់នៃចរន្តនិងតង់ស្រួល

ក្សួយសីបប្រកែទម្លៃ: មានសញ្ញាសំគាល់ដែលប្រាប់ពីពេលដោយត្រូវបានដូចជាដែលមានចរន្តដើរ ។

- ✓ អត្រាចរន្ត
- ✓ អត្រាចង់ស្រួល
- ✓ រយៈពេល-ចរន្ត (លេវ្តិននៃការការកាត់ដាច់)

- ✓ ក្រុមហ៊ុនផលិត
- ✓ វិញ្ញាបណ្ឌបត្រដែលទទួលស្ថាល់ដោយច្បាក់ជាតិ និងអន្តរជាតិ
- ✓ Breaking Capacity



រូបរាង ៥ ៖ ភ្លើយសុំបមានសញ្ញាសំគាល់ប្រើប្រាស់

៣.២. អង្វែងខ្លួន |_m

តីជាចរន្តអតិបរមាជែលមុននឹងភ្លើយសុំបត្របានកាត់ជ្លាច់ទាំងស្រីង ។

តារាង១៖ អត្រាចរន្ត និងទំហំខ្លួនដែង

អត្រាចរន្ត (A)	អង្គត់ជ្លាច់ខ្លួនដែនខាងក្រុងភ្លើយសុំប (mm)
3	0.15
5	0.20
10	0.35
15	0.50
20	0.60
25	0.75
30	0.85
45	1.25
60	1.53
80	1.8
100	2.0

៣.៣. លេវ្តីន

គីជារយៈពេលនៃការការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមពេលមានលំហែរចរន្តដើម្បីសលប់។ លេវ្តីនអាស៊យនឹងប្រភពរូបធាតុនៃខ្សោត្រប់។ លេវ្តីនដែលលេវ្តីនបំផុតគឺ 0.1 វិនាទី សម្រាប់ក្នុយសុំបន្ថែមលេវ្តីននិងចន្លោះពេល 10 វិនាទីសម្រាប់លេវ្តីនយើត។

៣.៤. អ្វតា I^2

អត្រានេះគីជាបរិមាណធាមពលហ្មរដ្ឋដែលការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមពេលដែលវាគ្រោរការកាត់ផ្ទាញ។ បន្ទាប់ពេលដែលការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមពេលវាគ្រោរការកាត់ផ្ទាញ។ អត្រាចាំងនេះប្រើបានកំណត់នៅក្នុងកសារបច្ចេកទេស (Datasheet) របស់ក្រុមហ៊ុនដែល ដលិត ក្នុយសុំបន្ថែម។

៣.៥. អ្វតាព័ត៌ម្ភ

អត្រាព័ត៌ម្ភបានកំណត់ត្រារៀង់ប្រើបានបុងប្រើបានការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមពេលមានលំហែរចរន្ត (Open-circuit –Voltage)។ ឧបាទរណ៍ យើងដោរក្នុយសុំបន្ថែមពេលមានអត្រា 250V ដើម្បីការកាត់ផ្ទាញចរន្តដែលហ្ម ចេញពីប្រភពដែលមានតម្លៃ 220 V។ ប្រសិនបើយើងយកកម្រិតក្នុងជានេះ ឧបាទរណ៍ 110V នោះនឹងមានការរែះខ្សោនក្នុយសុំបន្ថែម ហើយវានឹងការកាត់ផ្ទាញខ្សោយ។

៤. ទន្លោកំណត់ស្ថុរោង

ទន្លោកំណត់ស្ថុរោងនៃក្នុយសុំបន្ថែមទៅទៅកំណត់ដោយក្រុមហ៊ុនដលិត។ នៅពេលដែលមានចរន្តហ្មរកាត់ក្នុយសុំបន្ថែមស្ថុរោងនឹងការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមបានប៉ែងកែវឡើងក្នុងអត្រាបែរទៅតាមស្ថុរោងដោយតុកដូរការបស់វា។ នឹងក្នុងនឹងការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមនៃពេលវាយនាមដល់កម្រិតលីនីអី។

៥. ការកែងការនៃក្នុយសុំបន្ថែម

ក្នុយសុំបន្ថែមត្រូវការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែម។ ឧបាទរណ៍ក្នុយសុំបន្ថែមដែលមានអត្រា 1A / 25°C នឹងការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែមបន្ថែមមានពី 10% ទៅ 20% បន្ថែមឡើងនៅពេលប្រើបាន។ នៅស្ថុរោងការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែម 40 °C ប្រើបានការកាត់ផ្ទាញរបស់ក្នុយសុំបន្ថែម 80% នៃអត្រាបន្ថែម 100°C។

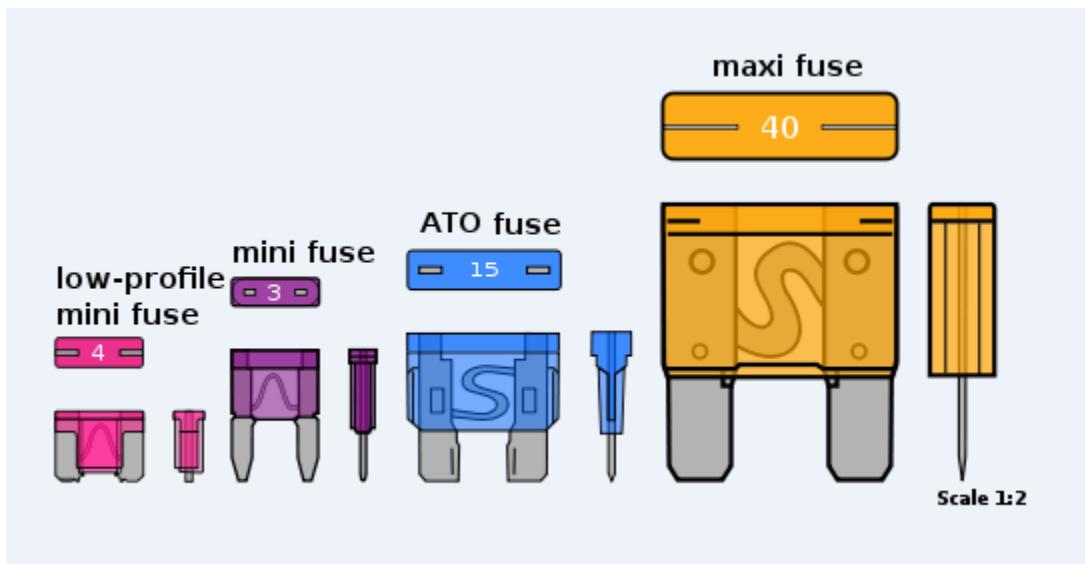
៦. ភ្លួយសីបយានយន្ត

ភ្លួយសីបយានយន្តប្រើសម្រាប់ការពារការធោះនៃខ្សែចម្លង និងខែករណីក្នុងយានយន្ត។

ភ្លួយសីបនេះមានច្រើនប្រភេទអារ៉ាស៊ីនិងការប្រើប្រាស់។ គេចាត់ថ្វាក់រាជាប្រឈម តី៖

- Blade Fuses
- Glass Tube
- Fusible Links
- Fuse Limiters

ការប្រើប្រាស់ភ្លួយសីបយានយន្តមានអត្ថាគន្លែង 32V ហើយប្រើក្នុងកម្រិតកន្លែង 24V។



រូបភាព ៦ : ភ្លួយសីបយានយន្ត

ថែរូលទី៤៖ កុលតាត់ (Switch)



រូបភាព ១ ៖ រូបពិតនៃប្រភេទកុលតាត់ដើម្បី។

១. សាយកសិរី

កុលសេវ្ទីអគ្គិសនីប្លូសេវ្ទីអេឡិចត្រូនិច កុលតាត់ (Switch) គឺជាប្រព័ន្ធបង្កើតអគ្គិសនីដែលមានត្បូនាទី កាត់ផ្ទាប់សេវ្ទី កាត់ផ្ទាប់លំហូរចរន៍ បូបដៃប្រឡាតិខ្សោយឬខ្សោយឡើតកុលសេវ្ទី ។ កុលតាត់មានប្រឹកសាខាប្រភេទ ១ ប្រភេទខ្លះ: អាចបញ្ហាដោយដែងផ្ទាល់ ប្រភេទខ្លះ: ឡើតអាចបញ្ហាដោយចលនា ដូចជាថ្នាបិទហើរ បញ្ហា ដោយសិកុណ្ណាកាត បញ្ហាដោយសម្ភារ បូលំហូរនៃវគ្គការ ជាដើម ។

២. ភាពខ្លាត់

ជាមួលដ្ឋានកុលតាត់មានដើងពត្តាប់អគ្គិសនីពីលោហេ: ពីរ គេហោចា Contact ។ កាលណាបានដើងទាំងពីរបែបដើរសេវ្ទីបិទ ហើយកាលណាបានដាច់ពីត្រាបោចាសេវ្ទីចំហេ ។ Contact ត្រូវមានភាពធមុន ទៅនឹងការប្រែ: ពីរបែបដើរសេវ្ទីបិទ ហើយកាលណាបានដើងទាំងពីរបែបដើរសេវ្ទីបិទ ។

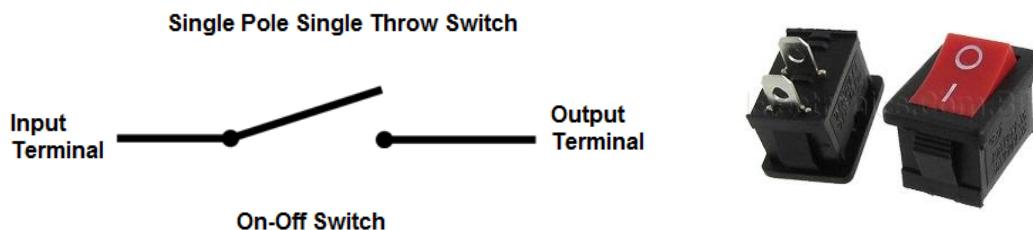
២.១. ឯកចាន់នៃភាពខ្លាត់ (Contact Terminology)

កុលអេឡិចត្រូនិច កុលតាត់ត្រូវបានបែងចែកទៅតាមលក្ខខណៈ: នៃការពត្តាប់ ។ Contact មួយគូរគេចាត់ទុកជាបិទ (Closed) នៅពេលដែលមានចរន៍ហូរឆ្លងការតែត្រូវក្រោមក្រិត ពង្រីក ដែលបានផ្តល់ឱ្យ ។ ភាពខ្លាត់បិទ (Open) នៅពេលដែលចរន៍ចំហេ ហើយត្រូវបានចរន៍ហូរឆ្លងការតែត្រូវក្រោមក្រិត ដែលបានផ្តល់ពង្រីក ។ គេនិយមប្រើពាក្យ “ Make ” ជីនុសពាក្យបិទ និងពាក្យ “break” (ការផ្តល់) ជីនុសចូរពាក្យចំហរបូហើរ ។

ចំណួន Poles របស់កុងតាក់កំណត់ចំណួនសៀវភៅផ្សេង ឬគ្មានដែលកុងតាក់អាចបញ្ជាណ ឬខាងក្រោមកុងតាក់ដែលមានមួយ Pole អាចបញ្ជាលេសៀវភៅបិទ បុរីកបានពេលចូលបញ្ជាណ កុងតាក់មានបីន Pole អាចបញ្ជាលេសៀវភៅបីនផ្សេង ឬគ្មាន ឬបិទបុរីក ។

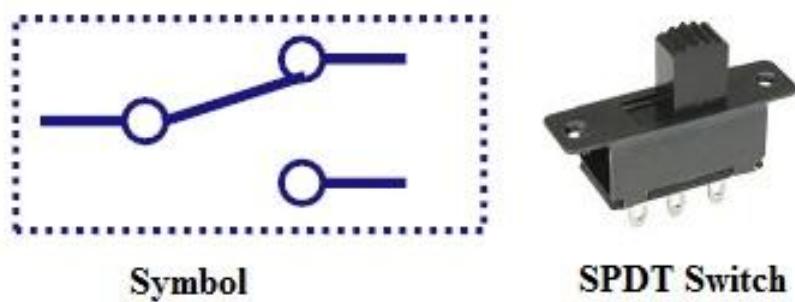
ចំណួន Throw កំណត់ពីចំណួនទីតាំង បុរីកលាក់ដែលកុងតាក់អាចភ្លាប់ ឬខាងក្រោមកុងតាក់ដែលមាន 1 Pole និង 2 Throw នៅលើមួយនៅអាចការចូលបញ្ជាណបាន កុងចំណោមចូលបញ្ជាណ ពីរ ។

- **SPST (Single Pole Single Throw)** មានធីជីវិត និងមុខងារភ្លាប់រវាងចូលបញ្ជាណ នៃសៀវភៅ ឬភាគច្រើនប្រើសម្រាប់បិទបុរីក ជាពិសេសបិទបុរីកសៀវភៅអគ្គិសនី ។



រូបភាព ២ : និមិត្តសញ្ញានៃ SPST និងរូបពិត SPST Rocker Switch

- **SPDT(Single Pole Double Throw)** ជាកុងតាក់សម្រាប់ដែលមានធីបី ឬកុងនៅលើ មានធីមួយជាភាគធីយុម (COM) និងធីទៅត្រួតជាភាគធីផ្ទាស់បូរ ។



រូបភាព ៣ : និមិត្តសញ្ញានៃ SPDT និងរូបពិត SPDT Slide Switch

- DPDT (Double Pole Double Throw) មានជីងុយ បង្កើតឡើងដោយ SPDT
ចំនួន ២ ។ ជាជម្រើសភាពអាចបញ្ជាសៀវភៅចំនួនពីរ ។



រូបភាព ៥ និមិត្តសញ្ញានៃ DPDT និងរូបពិត DPDT Rocker Switch

- 4PDT គឺមាន 4 Pole និង 2 Throw



រូបភាសា ៥ និមិត្តសញ្ញានៃ 4PDT និងរូបពិត 4PDT Toggle Switch

២.២. គារឆ្លងត្រឹម (Electrical Arc)

នៅពេលដែលអនុភាពដែលផ្តល់ទ្វាត់ ឬឡើចត្រូនដែលស្ថិតនៅចុងទាំងពីរនៃ Contact អាចមានលទ្ធភាពធ្វើអីយុងកម្ម (ionize) ឡើលើមួយលេខតុលខ្សោយដែលស្ថិតនៅចន្ទោះ Contact បង្កើត បានជាបន្ថែមជាស្ថាស្ថាគេហែថាការឆ្លងត្រឹម (Electrical Arc) ។ ផ្តាស្ថា (Plasma) មានសុវត្ថិភាព ដូចនេះ វាអាចឲ្យមានលំហ្អចរន្ត ឡារៈបីជាចន្ទោះរវាង Contact ចាប់ធ្វើមួយពីភ្ញាក់ដោយ ផ្តាស្ថាមានកំដៅក្នុង និងអាចធ្វើឲ្យច្រេះ Contact ហើយគុណភាពនៃកុងតាក់ចាប់ធ្វើមួយក្នុងចុងទាំងពីរនៃ Contact ។

៣. Actuator

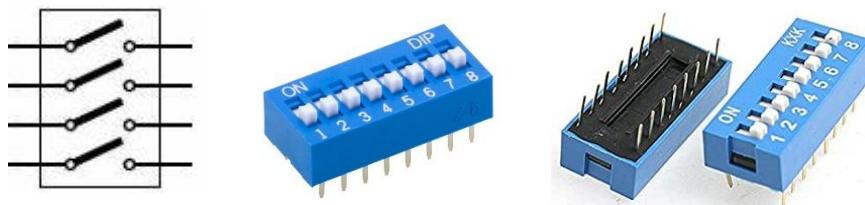
ផ្ទើកចលប់តដែលធ្វើឲ្យកុងតាក់ត្បាប់ត្រាគេហែថា Actuator ។ Actuator អាចជា Toggle Rocker ឬ Push Button បុត្រិណាប់មេកានិចផ្សេងៗទៀត ។

Actuator បីដែលបានប្រើប្រាស់ឡើងមានពីរភ៍៖ momentary និង Latched ។ momentary គឺជាប្រភេទដែលត្បាប់ឡើងប្រចាំម៉ោងសង្គមតែពេលប្រើប្រាស់ និងត្រូវបានបញ្ចប់ឡើងវិញ ។ Latched ជាប្រភេទដែលត្បាប់បានបញ្ចប់ឡើងសង្គមតែប្រចាំពេលដែលយើងបានប្រើប្រាស់ឡើង រាល់នៅក្នុងសាន្តភាពដែលជាបុចជាបុច ឬជាបុចជាបុច ។

៤. គុណភាពផ្សេងៗនៃ

៤.១. DIP Switch

ជាកុងតាក់ដែលមានស្ថិតិជាតិ Dual-in-Line ។ វាបានប្រភេទ SPST គឺប្រើបានជាដុំលើសៀវភៅ Digital ។



រូបភាព ៦ និមិត្តសញ្ញាណនូវបុព្ទនៃ DIP Switch

៤.២. Toggle Switch

ជាកុងតាក់ដែលមានប្រើប្រាស់ឡើងទំហំ ប្រើប្រាស់ប្រើប្រាស់ Pole និងប្រើប្រាស់ Throw ។



រូបភាព ៧ និមិត្តសញ្ញាណនូវបុព្ទនៃ Toggle Switch

៤.៣. Reed Switch

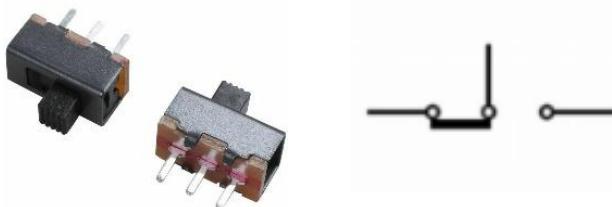
កុងតាក់ប្រភេទនេះដំណើរការដោយដែនម៉ាញ្ញីច។ នៅពេលដែលត្រានដែនម៉ាញ្ញីចផ្តួចកាត់ភ័ត៌ហា ហើយពេលបិទនៅពេលមានដែនម៉ាញ្ញីចផ្តួចកាត់។



រូបភាព ៤៖ និមិត្តសញ្ញា និងរូបរិពិត Reed Switch

៤.៤. កុឡាតាក់វិជ្ជ (Slide Switch)

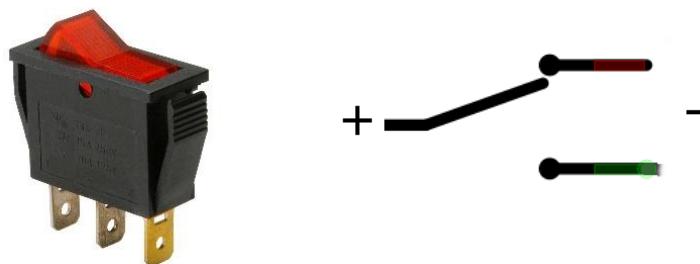
ជាកុងតាក់ដែលមាន ប្រើនៃ Pole និង ប្រើនៃ Throw ។ គេប្រើនៃប្រើរាល់នៅលើ PCB



រូបភាព ៥៖ រូបរិពិត របស់ Slide Switch និងនិមិត្តសញ្ញា

៤.៥. Rocker Switch

គេប្រើនៃប្រើសម្រាប់បិទ-បើកពង់ស្រួលចូល ។



រូបភាព ៦០៖ Rocker Switch ប្រភេទ Panel Mount មានអំពូលអ្នក 20A/125VAC
15A/250VAC និងនិមិត្តសញ្ញា

៥.៥. Limit Switch

គេអាចហេរិមារីងទេតិច Micro Switch ។ គេប្រើនឹងប្រើជាមួយប្រព័ន្ធមេកានិចខាងក្រោមណា ទាំងបុងណើរយណុ

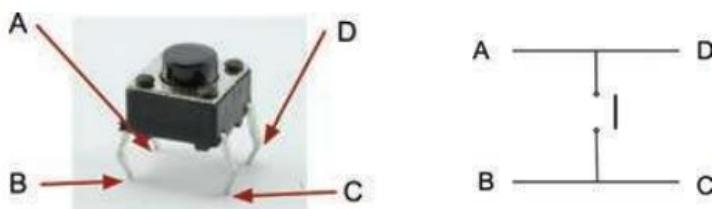
Limit switch symbols



រូបភាព ១១ : និមិត្តសញ្ញា និងរូបពិត limit Switch

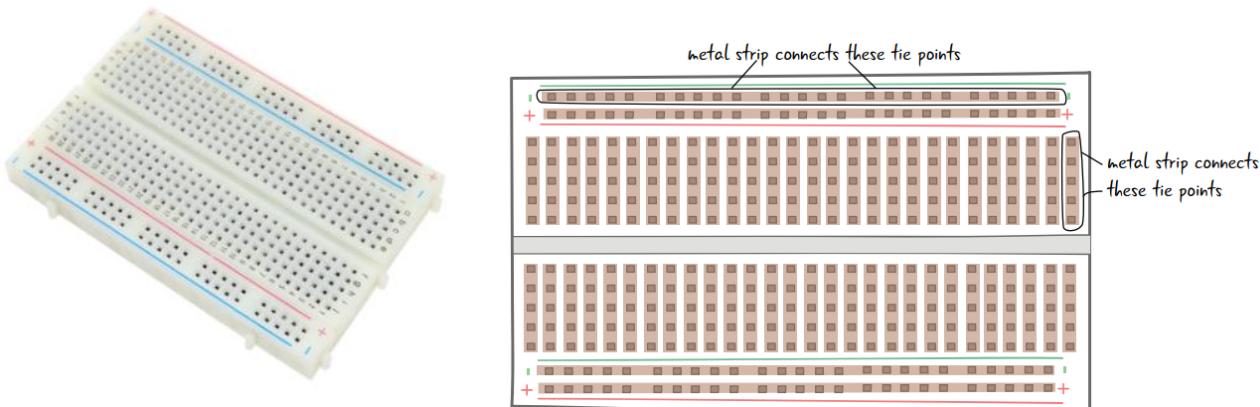
៥.៦. Push Button

Push Button មានតាមទំហំ និងទេតាមអត្រាអនុកាត ។ ប្រភេទដំឡើង គេប្រើនឹងប្រព័ន្ធដែលមានអនុកាតដំឡើងដូចតិចនៃលានសំណើម ។



រូបភាព ១២ : រូបពិត Push Buttonនិងនិមិត្តសញ្ញា

៥.៧. Bread Board



រូបភាព ១៣ : រូបពិត Bread Board

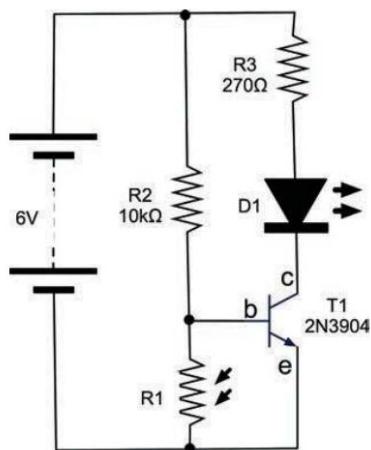
គំរូកលេខ៖ តិនោជន៍សេវាឌីអេឡិចត្រូនិកថាមតម្លៃ

តិនោជន៍ ១ ៖ LDR

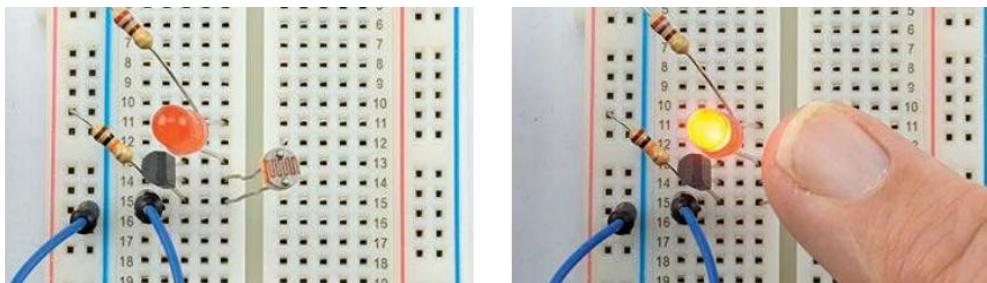
ក្រុមហ៊ុនសម្រាប់អនុវត្តន៍យោង

1. ប្រពិលមានទំហំ 6 V
2. ត្រង់សីស្ស់រ ប្រភេទ 2N3904 – 1 គ្រាប់
3. 330Ω resistor – 1 គ្រាប់
4. 270Ω – 1 គ្រាប់
5. LDR – 1 គ្រាប់
6. LED – 1 គ្រាប់
7. Jumper wires
8. Breadboard

សេវាឌីអេឡិចត្រូនិកត្រួតពិនិត្យ



របាយការងារ

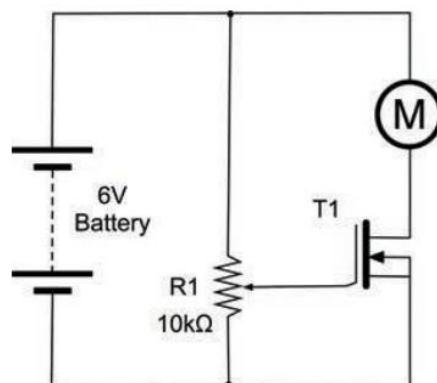


តិន្នន័យលេខ ២ នៃ លេខវិនិច្ឆ័យ Motor DC

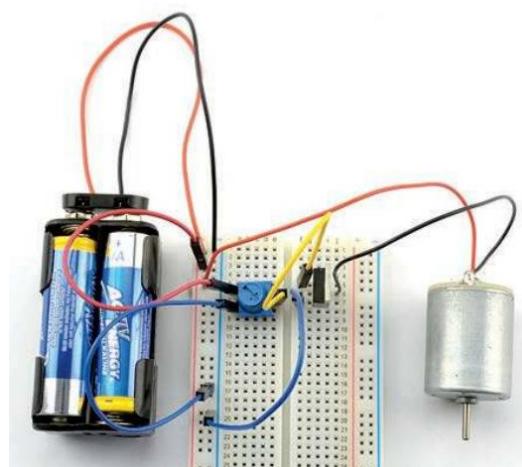
ក្រឡើតសម្រាប់អនុវត្តន៍ហាន

1. ប្រពិលមានទំហំ 6 V
2. MOSFET ប្រភេទ FQP30N06L – 1 គ្រាប់
3. 10kΩ resistor (នៃសីស្សដែលរីយ៉ាប់) – 1 គ្រាប់
4. Motor DC
5. Jumper wires
6. Breadboard

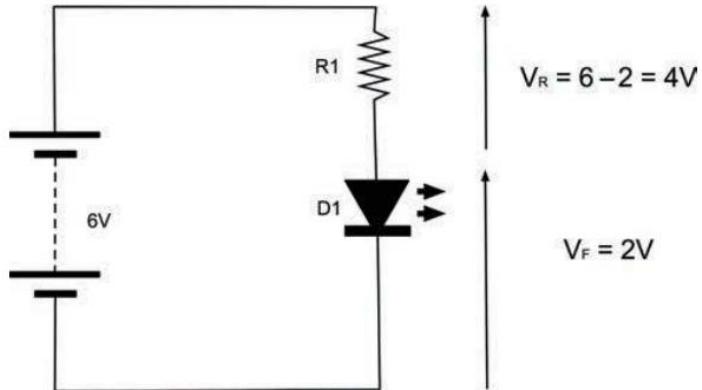
សេចក្តីផ្តើមនៃក្រឡើត



ក្រឡើតនៃក្រឡើត



គិតសោចន៍ ៣ ៖ គាយកតត់ឡើងតម្លៃ LED



$$V = IR$$

$$V = (0.020 \text{ amps}) * 220 \text{ ohms}$$

$$V = 4.4 \text{ Volts}$$

Supply Voltage (V)	Red	Green, Yellow, Orange	Blue
3	91Ω	60Ω	none
5	220Ω	180Ω	91Ω
6	270Ω / 330Ω	220Ω	180Ω
9	470Ω	470Ω	360Ω
12	680Ω	660Ω	560Ω

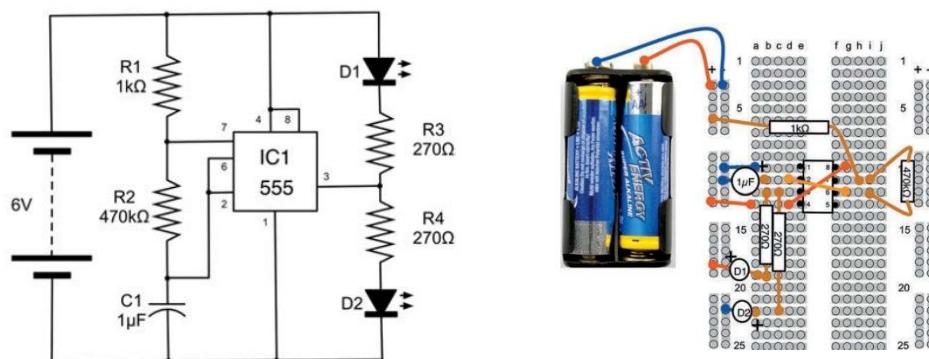
Parameter	Red	Green	Yellow	Orange	Blue	Units
Maximum forward current (I_F)	25	25	25	25	30	mA
Typical forward voltage (V_F)	1.7	2.1	2.1	2.1	3.6	V
Maximum forward voltage	2	3	3	3	4	V
Maximum reverse voltage	3	5	5	5	5	V

តិ៍នោជល់ ឯ៍ ៖ LED Blinking ដោយប្រើ IC555

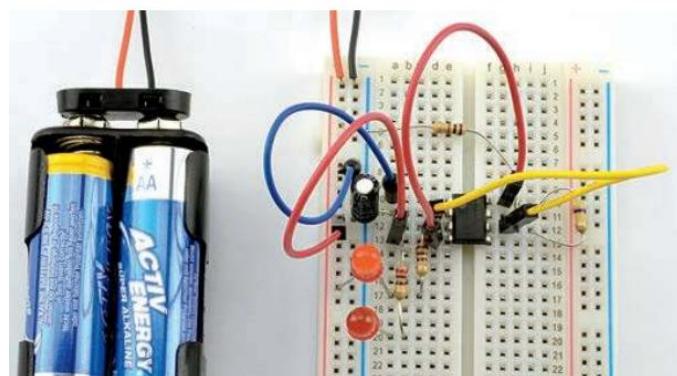
ក្រុមហ៊ុនប្រព័ន្ធគ្មាន

1. ប្រពិលមានទំហំ 6 V
2. IC555 – 1 គ្រាប់
3. 470Ω resistor – 2 គ្រាប់
4. 470kΩ resistor – 1 គ្រាប់
5. 1kΩ resistor – 1 គ្រាប់
6. LED – 2 គ្រាប់
7. កុងដឹងប្រកាន់បូល 1uF -1 គ្រាប់
8. ខ្សែសម្រាប់ធ្វើតាត (Jumper wires)
9. Breadboard

សេចក្តីផ្តល់នូវការ



របៀបធ្វើនា



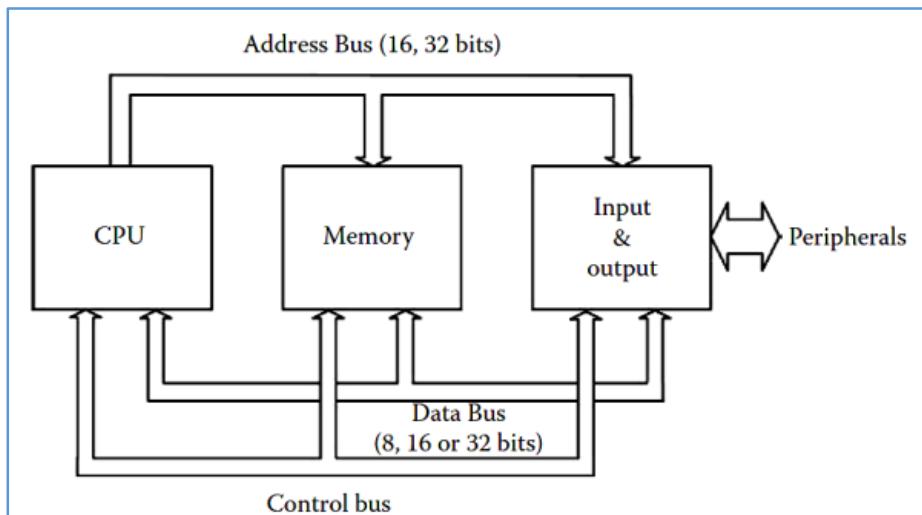
ថំពូក ៣ ៖ Microcontroller

នេះត្រូវលក្ខិណ៍ សញ្ញាណនៃ Microcontrollers

១. ធម៌ដែលគិត: នៃ Microcontrollers និង Microprocessors

Microcontroller គឺជាដឹកកញ្ចប់មួយនៃប្រព័ន្ធបញ្ហាដែលគេជាក់នៅទៅក្នុងបន្ទះយើហែតមួយ ។ ហើយវា មានសៀវភៅអេឡិចត្រូនិចជាប្រើប្រាស់ដែលបានបង្កើតសម្រាប់ការបញ្ហា និងបំផុែងរាយជាសញ្ញា អគ្គិសនី។ ជាទុទៅ microcontroller គឺជាស្ថុលដំណើរការសម្រាប់ការបញ្ហា ការចងចាំ, ដំណើរការ ច្បាស់លាស់ ហើយគ្រឿងរបស់វា មាននៅលើបន្ទះយើហែ, ហើយវាត្រូវដំណើរការបានត្រូវកំណត់មានកម្មវិធី សម្រាប់គ្រប់គ្រងរាយ។

ឧាជក្រាមបង្ហាញពីបុកដ្ឋានក្នុងក្រុមក្រាមគឺមួយនៃ Microcontroller ។ នៅក្នុងនោះវាដឹងដោយបុក សំខាន់ៗ បីគី Central Processing Unit (CPU), អង្គចងចាំ(Memory) និងប្រភពចូល&ប្រភពចេញ (Input & Output(I/O)) ។



រូបភាព ១ ៖ បុកដ្ឋានក្នុងក្រុមក្រាមគឺមួយនៃ Microcomputer (CPU គឺជាវិធី Microprocessors)

បុកនឹងមួយនៃគ្រប់បានតាមបញ្ហាដោយបណ្តុះនៃខ្សោចមួងអគ្គិសនី ហែងចា Bus ។ ហើយ Bus មានពីរប្រភេទគី Data Bus និងAddress Bus ហើយ Data Bus មានរូបនាទីដីកជាតុនទិន្នន័យដែល មានសីញ្ញាល់បញ្ហា (Control signal) ទៅលើAddress Bus ។ ចំពោះAddress Bus មានរូបនាទី ដីកជាតុនអាស៊យដ្ឋាននៃ Memory ឬ Input / Output (I/O) ។

CPU គឺជាទូរក្រុលនៃ Microcomputer ហើយបន្ថែមការបញ្ជាផោយកម្មវិធី ដែលរក្សាទុកនៅក្នុង Memory ហើយនៅក្នុង CPU មានបណ្តុះស្រួលពិស់សម្បូយហេរិចា Arithmetic and Logic Unit (ALU) ។ ALU មានមុខងារធ្វើប្រមាណវិធីនៅត្រឡប់ និង logic gate ដាម្បូយការសារបស់ ម៉ាសីនប្រព័ន្ធគោលពីរ (Binary number system) ។

CPU គឺជាកម្រិត Microprocessors របស់ Microcomputer ដែលមានតួនាទីប្រតិបត្តិនូវរាល់កិច្ចការ ដូចដែលបានរៀបរាប់ខាងលើ ។ Microcontroller ត្រូវបានគេសន្តែតាតា Microprocessors ដែលបង្កើតឡើងនៅលើអីសម្បូយ បុបន្ទះលើឬ ។ Microcontroller គើតឡើងក្រោយ Microprocessors ហើយលើរលើគ្នានឹងខុសត្រូវ ។ គេប្រើប្រាស់ Microprocessors នៅក្នុងម៉ាសីនកំពុំរដ្ឋាល់ខ្ពស់ បុ Workstation ពីព្រោះវាគ្រូវការអនុវត្តន៍ការគណនាចាំ មានសមមត្តភាពអាចត្រួតព្រឹងសំន឴និនូវនិងពាក្យបញ្ហាដីប្រើប្រាស់ ក្នុងលេវ្កិនលើវីន ។ ចាត់វាមេត្រមួយដីសំខាន់នៃ Microprocessors គឺទំហំនៃចិត្តទិន្នន័យ (Registers) ខាងក្រុងរបស់វាមាន 8, 16, 32, 64 bits ដែល CPU អាចយកទៅដំណើរការក្នុងមួយលើកែ ។ Microcontroller ត្រូវបានគេយកទៅអនុវត្តក្នុងវិស់យោជាប្រើប្រាស់មានដូចជា ៖ ឧស្សាហកម្មរចយន សម្រារ៖ ទូរគមនាគមន៍ សម្រារ៖ ប្រើប្រាស់ក្នុងផ្ទះ សម្រារ៖ ពិសោធន៍បរិភាពទេរ ឧបករណ៍កំសាន្ត ។ លើ ។ គេប្រើប្រាស់ Microcontroller ក្នុងកិច្ចការណាដែលមានមុខងារសមញ្ញ តិចតុច និងមានតម្លៃថាកតាមដែលអាចធ្វើទៅបាន ។ ជាទុទេវាដំណើរការដោយស្ម័គ្រប់ត្រួតព្រឹងកម្មវិធីដែលបានរក្សាទុកនៅក្នុង Memory ហើយដើងចេញចូលរបស់វាមានមុខងារទៅកាត់ទំនងទៅនឹងប្រភពក្រោ ។ ចំពោះកិច្ចការដែលសំញ្ញាំគេអាចប្រើ Microcontroller ប្រើប្រាស់ក្នុងប្រព័ន្ធ តែមួយបាន ។

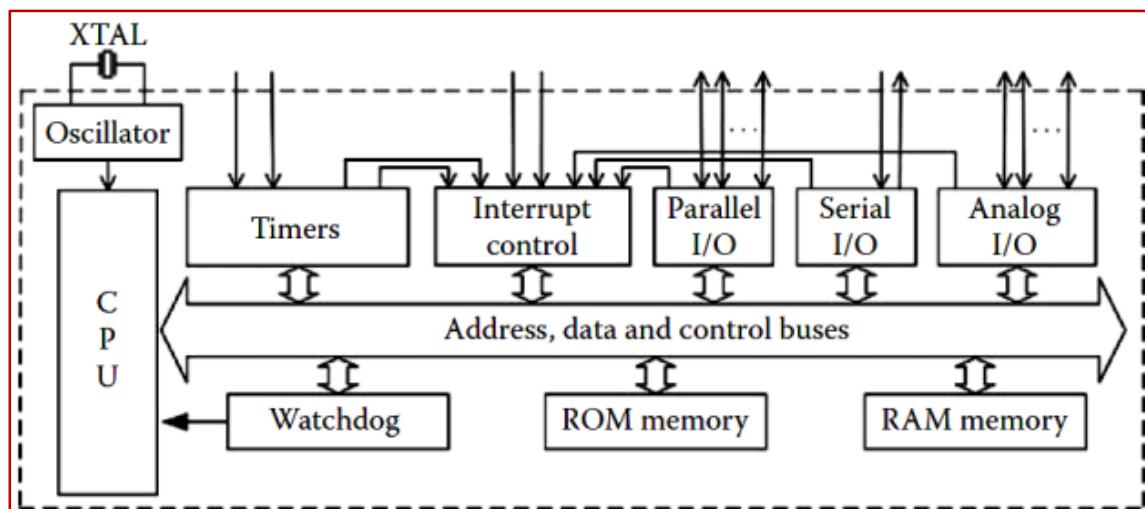
ខាងក្រោមគឺជាគន្លឹះគម្រោះក្រោរការចាំបាច់ទូទៅនៃ Microcontroller ដែលត្រូវយកមកប្រើប្រាស់៖

1. ធ្វើក្រួចចូលនិងក្រួចចូល (Input/Output) ៖ ធ្វើក្រួចចូលរបស់វាគ្រូមានមុខងារធ្វើដោយក្នុងខុសវគ្គប្រើប្រាស់ ដើម្បីបង្កើតជាអាជីវការ និងក្រួចចូលដែលបានបង្កើតឡើងដោយក្រួចចូលរបស់វាមានមុខងារធ្វើដោយក្នុងខុសវគ្គប្រើប្រាស់ ។ គេអាចធ្វើក្រួចចូលរបស់វាមានមុខងារធ្វើដោយក្នុងខុសវគ្គប្រើប្រាស់ ។
2. មានទំហំការងារកំពុច ៖ ត្រូវមានទំហំក្នុចហើងតម្លៃសមរម្យ ។ ចំនួនធ្វើការស្រែយទេនឹង រូបភាព ខាងក្រោររបស់អីស់ ។ គេអាចធ្វើអោយការកំពុចដោយអោយដើងនិមួយ ។ មានមុខងារធ្វើដោយក្នុងខុសវគ្គប្រើប្រាស់ ។

3. ប្រើប្រាស់អោយសាកសម្បទៅនឹងកិច្ចការ៖ នៅក្នុងអំបូរៈនៃ Microcontroller មួយមានចំណាំ Memory ចំនួនដើម្បីចែងច្នៃល ខុសទៅគ្នា បើនេះសំនួរក្រុមហ៊ុន និងការសរស់រក្សាទីដើម្បីធ្វើការ ។ ដូចនេះយើងប្រើសិសពីសំនួរក្រុមហ៊ុន Microcontroller ទៅតាមចំណាំនឹងសមមត្ថភាពដែលយើងចង់បានក្នុងអំបូរៈនោះដោយប្រើកម្មវិធីដែល (ប្រាការកំណត់ប្រព័ន្ធឌីតុល) ។
4. ដំណើរការមានស្ថារភាព៖ យើងត្រូវធានាថា Microcontroller ដំណើរការកម្មវិធីបានត្រឹមត្រូវដោយគ្មានការរំនៅស្ថៃ ប្រើលើងដំណើរការ ។ ជាមួយ Microcontroller បំពាក់ទៅដោយ Watch Dog Timer (WDT) ដើម្បីធានានូវដំណើរការត្រឹមត្រូវ ។
5. ប្រើប្រាស់ចាមពលពិចេះ ចំពោះប្រព័ន្ធដំណើរការដោយប្រើកិច្ចពិលគេត្រូវប្រើ Microcontroller ដែលសុំចាមពលពិចេះ ។ ជាទុទេ Microcontroller សុំចាមពលពិចេះបំផុតនៅពេលដែលវាស្ថិតក្នុងលក្ខណៈស្រួច (Standby ឬ sleep-mood) ហើយនេះជាប្រើកិត្តិការណាតីខាងក្រោមដើម្បីដំណើរការ ។
6. ការពារពីការលួចចែកចែង៖ កម្មវិធីដែលបានរក្សាទុកនៅក្នុង Memory ត្រូវតមានសុវត្ថភាពពីការអាងទិន្នន័យប្រើការចែកចែងដោយជនិកមួយដោយគ្មានការអនុញ្ញាតពីមាតស់កម្មសិទ្ធិ ។

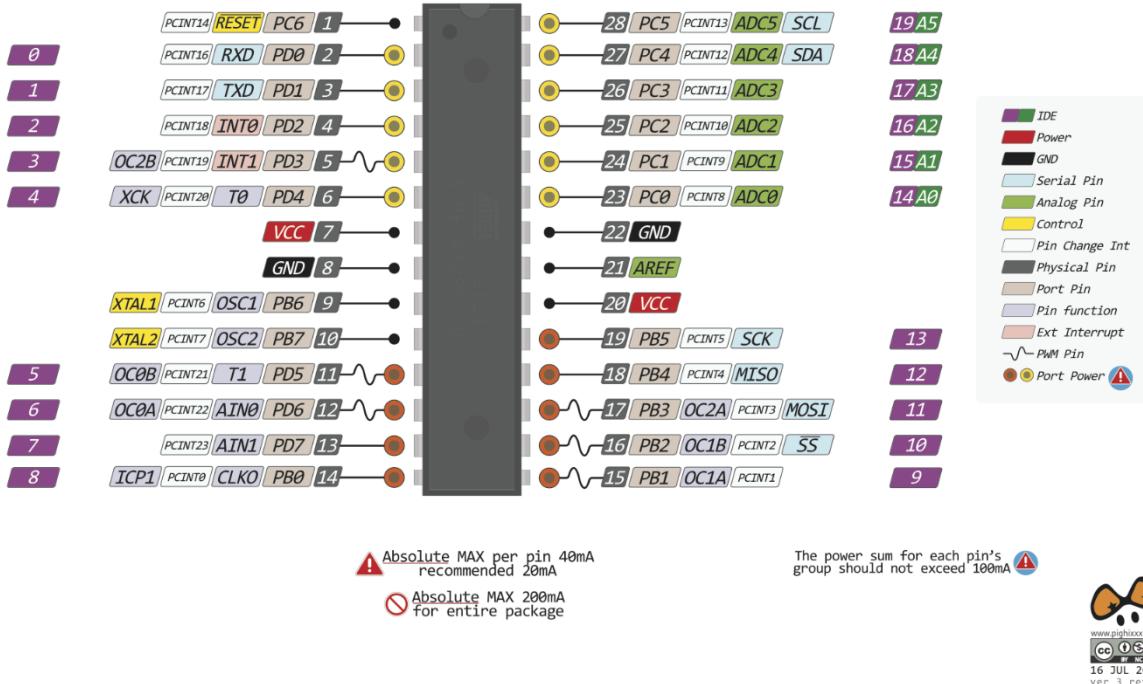
២. ឧត្តម្មីនៃ Microcontroller

Microcontroller បានឱ្យប្រើប្រាស់ដោយជាតុកធម្មសំខាន់របស់ Microcomputer ដូចជា CPU , Memory , I/O នៅលើបច្ចេកទេសកុម្ភិភាព ។



រូបថត ២ ៖ ប្រព័ន្ធឌីតុលនៃ Microcontroller

ATMEGA328 PINOUT



រូបភាព ៣ : Microcontroller ប្រភេទ ATMEGA328P

- ❖ **អូសិលីឡូ (Oscillator) :** បង្កើតសីញ្ញាល់ដើម្បីសាំងត្រនកល់ប្រពិបត្តិការខាងក្រុងរបស់ Microcontroller ។ ការប្រើអូសិលីឡូមានរចនាឌម្រិះមានផ្ទុចធាតាការប្រើប្រាស់តែនសិស្សដែលត្រួតពិនិត្យការងារ។ និងកងដង (RC) បូប្រើបញ្ជីកត្រីស្ថាល់ (quartz Crystal or XTAL) ។ ជាទុទេតែប្រើបញ្ជីកត្រីស្ថាល់ត្រូវបានស្វែងរកដោយស្រួលប្រើប្រាស់និងដំណើរការមានស្អែរភាព និងកម្រិតលម្អិត។

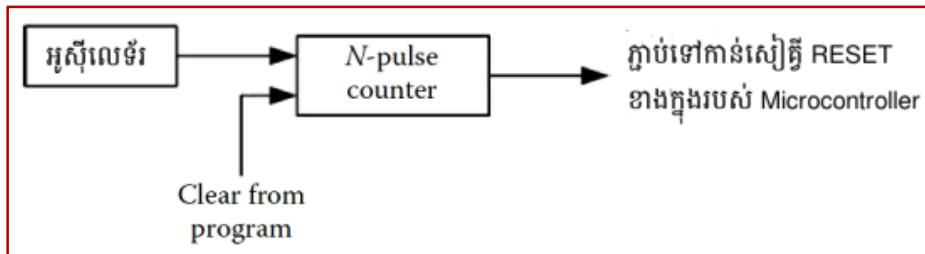


រូបភាព ៤ : ប្រភេទដៃងគ្មានត្រីស្ថាល់

- ❖ **CPU :** គឺជាទូរក្រុល (Processor Core) សម្រាប់បញ្ជា រាយការសម្រាប់ការគណនោយ លើអ្នកសរស់រកម្មវិធីដោយជំណើរការធ្វើអ្នកម្មយ ។
- ❖ **Memory :** គ្រប់Microcontrollerទាំងអស់សុទ្ធផ័ត៌មានអង្គចងចាំ(Memory)ដើម្បីរក្សាទុកទិន្នន័យ នៃពាក្យបញ្ជា ។ Microcontroller មានMemory ពីរប្រភេទគឺ Random Access Memory (RAM) និង Read Only Memory (ROM) ។ ទិន្នន័យនៃក្នុង RAM អាចបើកមើល (Read) និងកែប្រែ (Write) បាន ។ RAM អាចរក្សាទុកទិន្នន័យដោយត្រូវការអង់ស្សាន ហើយទិន្នន័យ នឹងបាត់បង់ឡើង នៅពេលត្រួតពិនិត្យ តង់ស្សានផ្តើមតែង ។ ទិន្នន័យនៃក្នុង ROM បានត្រឹមតែបើកមើលប៉ុណ្ណោះ (read) ហើយវាចិនត្រូវការ អង់ស្សានដើម្បីរក្សាទុកទិន្នន័យឡើង ។ បច្ចេកវិទ្យា ធ្វើនៅក្នុងដែលគេប្រើប្រាស់សម្រាប់ដំលិត ROM មាន ផ្ទុចជា EPROM (Erasable Programmable Read Only Memory) , EEPROM(Electrical Erasable Programmable Read Only Memory) , OTP (One Time Programmable) និង Flash ។ ទាំង ROM និង RAM គេហេចថាគារប្រើប្រាស់បានប្រើប្រាស់ប្រព័ន្ធឌីតុលិក “Random Access” ពីរបារាំង ប្រើប្រាស់ទិន្នន័យ នៅខាងក្រោមរបស់វាដោយមិនអាស់យោគឡើង នៅទីតាំងបុរាណសំយដ្ឋានដែលរក្សាទុកទិន្នន័យនៃក្នុង Memory ទេ ។

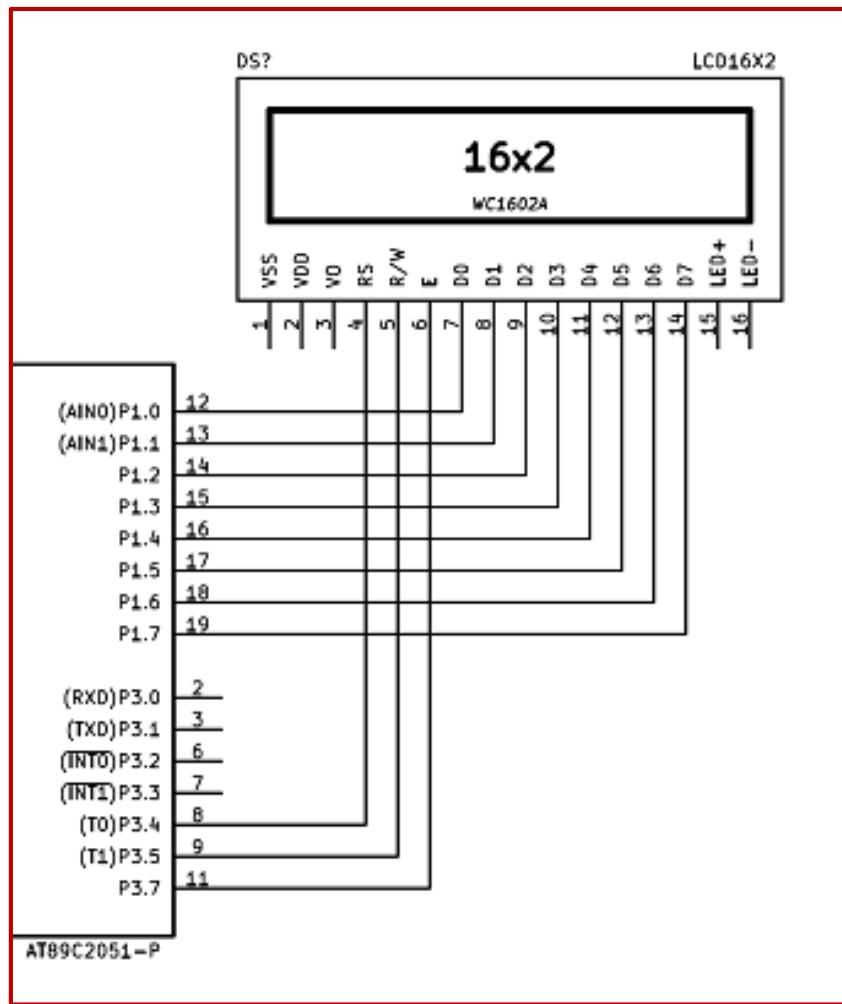
ROM ត្រូវបានប្រើប្រាស់ដើម្បីរក្សាទុកម្មវិធីអចិន្តូយ៉ែត RAM ត្រូវបានប្រើប្រាស់ដើម្បីរក្សាទុកទិន្នន័យដែលត្រូវបានប្រើប្រាស់ដោយកម្មវិធីជាបណ្តាល់អាសន្ន ។ Microcontroller ប្រភេទចិះ មួយចំនួនមានបន្ថែម EEPROM ដើម្បីរក្សាទុកទិន្នន័យបន្ថែម ។ EEPROM គឺជា Memory ដាច់ដោយ ឡើងអាចរក្សាទុកទិន្នន័យដោយមិនចាំបាច់មានអង់ស្សានផ្តើមតែង ។

- ❖ **Watch dog :** បង្កើតឡើងដោយអ្នកសិលទេរ និង Binary-Counter ទំហំ N Bits ហើយ Microcontroller ភាគច្រើនបំពាក់ឡើង watch dog Timer (WDT) ។



រូបភាព ៥ : ងារប្រាម៉ែន WDT (ប្រកបឡាបស់វាត្រូវបានត្រូវការបោះឆ្នែកនៃសៀវភៅ Reset ខាងក្រោម Microcontroller ដើម្បីបង្កើតសិលទេរ Reset)

- ❖ **I/O** : គឺជាដីដីដើម្បីផ្តល់សញ្ញាផ័ត៌នាទំនាក់ទំនងនិងប្រកបពាណិជ្ជកម្មខាងក្រោម ហើយអាចមានមុខងារប្រចាំថ្ងៃដូចខាងក្រោមដូចជា : Serial-Port , Parallel-Port , Timer , Interrupt , និងធីស៊ីនីដី: មានមុខងារពាក់ព័ន្ធសិក្សាល់អាណាពុកមានដូចជា: Analog-to-Digital Converter (A/D) ឬ Digital to Analog Converter (D/A) ។
- ❖ **Parallel-Ports** :បង្កើតឡើងដោយបណ្តុះខ្សោយ I/O ឌីជីថលចំនួន 8bits ។ ដែលការរំនៀកការបញ្ចូន ប្រទួលទិន្នន័យមានភាពតំណាងត្រាបូយើងភាថយកខ្សោយ I/O ណាមួយមកប្រើក់បានដែល ។

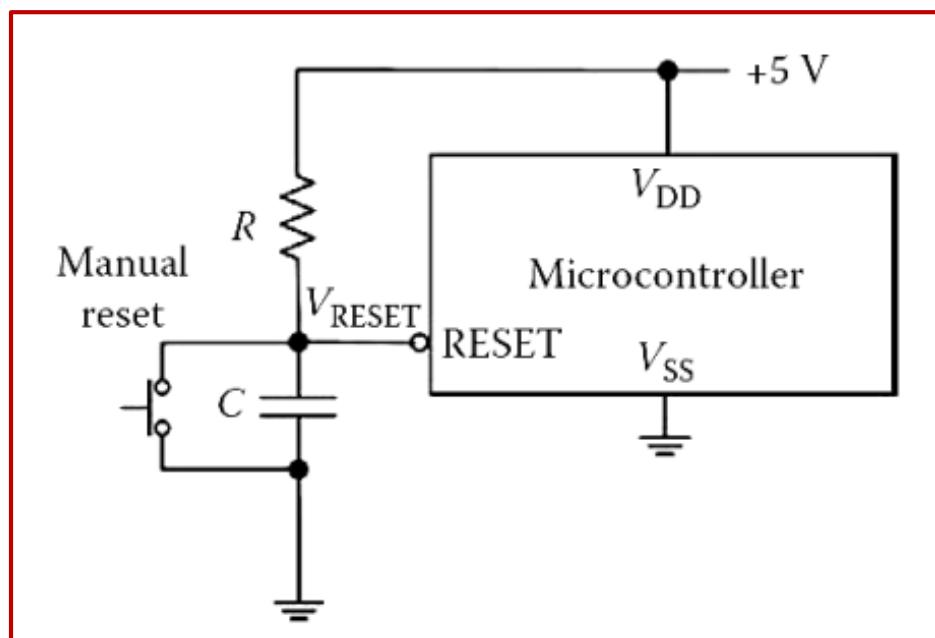


រូបនោះ ៦៖ ឧបាទរណ៍នៃការប្រើប្រាស់ Parallel-Port របស់ Microcontroller ដឹងរីបច្បាស LCD

- ❖ **Serial-Ports** : មានប្រភេទដើម្បីផ្តល់ព័ត៌មាន RS-232C (Recommended Standard 232, Revision C) I²C (Inter-integrated Circuit), USB (Universal Serial BUS) និង Ethernet ។

៣. ស្វែងរក្សា Reset

ដំណឹកការសារដើម (Reset) ធ្វើឱ្យ microcontroller ឬ microprocessor ត្រួលប់ទៅស្ថានភាពដើម ប្រចាំនុចចាប់ផ្តើមរបស់វាកិច្ច។ ដំណឹកការសារដើមកែតាអនឹងនៅពេលដើង Reset (Reset-Pin) ទួលលក្ខណៈ សីញ្ញាល់ Reset ។សីញ្ញាល់ Reset កំណត់ឱ្យ Program-Counter(PC) ត្រួលប់ទៅលក្ខណៈដើលបានកំណត់ ឧទាហរណី PC=0 ធ្វើឱ្យ microcontroller ឬ microprocessor ចាប់ផ្តើមប្រពិបត្តិកម្មវិធីពីភាស់យដ្ឋាន Memory ជាក់លាក់នោះ។ នៅក្នុង microcontroller គឺអាចបង្កើតសីញ្ញាល់ Reset ដោយចុចបុគ្គិន៍ Resetដើលភ្លាប់ ទៅនឹងដើង Reset របស់វា ឬ Reset នៅពេលដើលរាយបើកភ្លើងឱ្យដំណឹកការ (power-on-reset) ។



រូបភាព ៣៖ Reset ដោយប្រើបុគ្គិន៍ Reset ដើលភ្លាប់ ទៅដើង Reset របស់ microcontroller

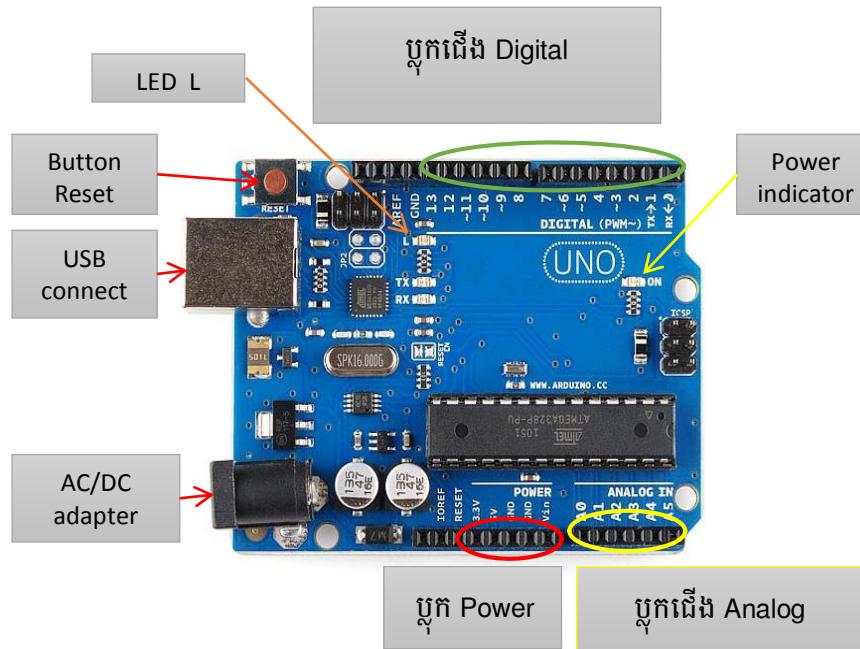
ផ្សេងៗទី២ ៖ ការពិចារណា និងការកែតាមការកែតាមការស្នើសុំ



រូបភាព ១ ៖ រូបតិនៃ Arduino UNO

១. ការកែតាមការស្នើសុំ និងការកែតាមការស្នើសុំ

Microcontroller	: ATmega 328P
Operating Voltage	: 5V
Input Voltage (recommended)	: 7-12V
Input Voltage (limits)	: 6-20V
Digital I/O Pins	: 14 (of which 6 provide PWM output)
Analog Input Pin	: 6
DC Current per I/O Pin	: 40 mA
DC Current for 3.3V Pin	: 50 mA
Flash Memory	: 32 KB
SRAM	: 2 KB
EEPROM	: 1KB
Clock Speed	: 16 MHz



រូបភាព ២៩: Arduino Uno Board

២. ផ្តល់ស្នើសុំតម្លៃ (POWER)

ប្លុកប្រកបដូតតែងតម្លៃ (Power) មានពីរប្រភពតី ដោយប្រើ AC-DC Adapter និងប្រភពតី +5V ពី USB ។ ហើយប្លុកPower នឹងព្យាយាយ VIN , 5V , 3.3V , GND ។

- ❖ **VIN** : ជាគង់ស្បែងដូលដោយ AC-DC Adapter ។ វាមានចន្ទោះពី 7V ទៅ 12V ប្រសិនបើលើស 12V វាអាចគ្វាន់ប្លុកប្រកបដូចខាងក្រោម។
- ❖ **5V** : ជាកម្រិតកង់ស្បែងដែលដោរការប្រើ ធ្វើបញ្ជីលើក្នុងUNO ។ ប្រភព 5V នេះបានមកពី VIN បុ កង់ស្បែងពី USB ។
- ❖ **3.3V** : ជាប្រភពកង់ស្បែងបង្កើតឡើងដោយ DC Converter នៅលើធ្វើបញ្ជីលើក្នុងUNO សម្រាប់ប្រើប្រាស់ខាងក្រោមនេះ។
- ❖ **GND**: គឺជាដើម្បីម៉ាស់ (Ground) រួម (ជីថិកតុត ប្រើបង្កើន 0V)

៣. ផ្តល់សិទ្ធិ (Digital Pin)

ប្លុកប្រើបង្កើនខ្លួនអាចបង្កើតប្រើបង្កើនប្រភពបញ្ហាប្រភព (Input/Output) បានតាមការកំណត់របស់អ្នកសរស់បញ្ហាទៅលើប្រើបង្កើននីមួយៗរបស់ក្រុង ។ ហើយប្រើបង្កើននីមួយៗរបស់ក្រុងការកំណត់របស់អ្នកសរស់បញ្ហាទៅលើប្រើបង្កើននីមួយៗរបស់ក្រុង

កម្រិតកដ្ឋាន +5V និងមានលំហែចរន្តចូលបាន 40 mA ។ ដើម្បីមួយមានសុស្សដៃខាងក្រោម (Internal pull-up resistor) ដែលត្រូវប៉ែនីង 5V ។ និងសុស្សដៃខាងក្រោមមានសុស្សដៃថ្វាម៉ោះពី 20 KΩ ទៅ 50 KΩ បើនេះជាទូទៅដើម្បីមួយរបស់ភាគសៀវភៅតីចំហា ។ លើសពីនេះទៅទៀតមានដើម្បីមួយចំនួនមានមុខរៀងរាល់ទៀតដូចជា៖

- ❖ **Serial** ៖ មានដើមលេខ 0 តី Rx (ទទួលសុគ្រាល់) និងដើមលេខ 1 តី Tx (បញ្ចូនសុគ្រាល់) រាធម៌ណីរការក្រុងកម្រិត ទូសុក ប្រភេទ TTL ។
- ❖ **Interrupt** ៖ ដើមទី 2 និងទី 3 ។ រាយការបង្កើតការបង្ហាក់ទៅ CPU នៅពេលដើម្បីមួយមានតម្លៃទាប (LOW) បុប្បប្បលពីទាបទៅខ្ពស់ និងពីខ្ពស់មកទាប ។
- ❖ **PWM (Pulse Width Modulation)** ៖ មានដើមទី 3, 5, 6, 9, 10 និងដើមទី 11 ។ PWM គឺជាការបង្កើតសុគ្រាល់អាណាពាណុកចេញពីដើមឱ្យឯកសារ ។ ការបង្កើតគីប្រើទៅលើបញ្ហា លើវិនិម័យរបស់មួយចរន្តជាប់ (Motor DC) រាយការ Resolution 8-bits (256រៀងរាល់) ។
- ❖ **SPI (Serial Peripheral Interface)** ៖ ដើមទី 10 (SS) ដើមទី 11 (MOSI) ដើមទី 12 (MISO) និងដើមទី 13 (SCK) ។
- ❖ **LED** ៖ ប្រើទៅលើដើមទី 13 ស្រាប់ ។ បើ LED រលក់ តីដើមទី 13 ជាតុល និង បើ LED ត្រូវ តីដើមទី 13 ជាតុល ។

៥. ផ្ទុកលើលក្ខណាថ្មីក (Analog Pin)

ប្រើប្រាស់លក្ខណាថ្មីកក្នុងការអាជីវកិច្ច UNO មានដើម្បីមួយចំនួនជាប្រាំមួយគីប្រើដើម្បីជួយការក្រុងកម្រិត A0 , A1, A2, A3, A4 និង A5 ។ ក្នុងដើម្បីមួយមាន Resolution ចំនួន 10-bits (មានតម្លៃខ្ពស់ចំនួន 1024) បែបឈរ មានដើម្បីមួយចំនួនមានមុខរៀងរាល់ទៀតដូចជា៖

- ❖ **I2C** ៖ ជាប្រភេទ TWI ដែលដើមទី A3 (SDA) និងដើមទី A4 (SCL) ។
- ❖ **AREF** ៖ គឺជាការដើម្បីក្រុងកម្រិតដែលបានបង្កើតឡើងជាប្រាំមួយគីប្រើដើម្បីការបង្កើតការសារជាប់។
- ❖ **RESET** ៖ គឺជាការដើម្បីប្រើប្រាស់ការចូលចេញពីការក្រុងកម្រិតដែលបានបង្កើតឡើង។ ពេលវាគីតាល និងកំណត់ទូរសព្ទអីដូចជាការសារជាប់ ។

៥. ទំនាក់ទំនង (Communication)

ក្នុង Arduino UNO មានទំនាក់ទំនងពីរគឺរវាង Arduino និង កំពូលទៅ និងរវាង Arduino និង Microcontroller ដើម្បីបញ្ជីពីរបៀបប្រព័ន្ធអ្នកដែលបានរៀបចំឡើង Arduino UNO មាន អីសែនពីរប្រព័ន្ធសម្រាប់ធ្វើការ ទំនាក់ទំនងគឺប្រព័ន្ធគ្រប់ ATmega8U2 និង ATMega328p ។

- ❖ អីសែន ATmega8U2 មានក្នុងនាទីជាស្ថានទាក់ទងរវាង TTL USART ពី ATMega328P ទៅនឹង USB port របស់កំពូលទៅ ។
- ❖ ក្នុង IC ATMega328P មានប្រព័ន្ធពីរប្រព័ន្ធដែលបានរៀបចំឡើង ទំនាក់ទំនង TTL USART និង TWI (SPI និង I2C) ។

៦. ផ្ទាល់ខ្លួន (Memory)

នៅខាងក្រុង IC ATMega328P មានអង្គចងចាំបីខ្លួនគ្នា តី៖

- ❖ **Flash Memory** : សម្រាប់រក្សាទុកក្នុង (កម្មវិធី) មានទំហំ 32kB ។ ក្នុងនោះ Bootloader ត្រូវការផ្តល់នៅ Flash ប្រមាណ 0.5kB ។
- ❖ **SRAM** : សម្រាប់រក្សាទុកទិន្នន័យបណ្តុះបណ្តាលកម្មវិធីក្នុងអីដូចដែលការងារ។ វាបានទំហំ 2KB ។
- ❖ **EEPROM** : គឺជាកំណត់រក្សាទុកទិន្នន័យបណ្តុះបណ្តាលកម្មវិធីក្នុងអីដូចដែលការងារ។ វាបានទំហំ 1kB ។

ទេសចរណ៍ តាមលក្ខណៈការបង្កើតកម្មវិធី និងការព្រឹមប្រាក់ក្នុងបច្ចុប្បន្ន

១. ការបង្កើតកម្មវិធី Arduino

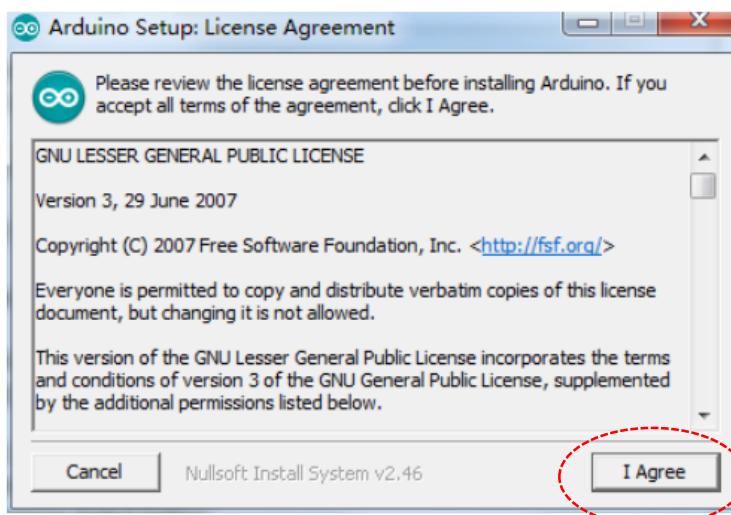
ដើម្បីទាញយកកម្មវិធី Arduino IDE យកមកប្រើប្រាស់ក្នុងកំពូទ័រអ្នកត្រូវចូលទៅកាន់គេហទំនាក់របស់ Arduino ដែលមានលេខា៖ថា <https://www.arduino.cc/en/Main/Software> បន្ទាប់មកទាញយកកំណែចុងក្រាយបង្គស់ ហើយតាំងឡើងនៅលើកំពូទ័របស់អ្នក ។



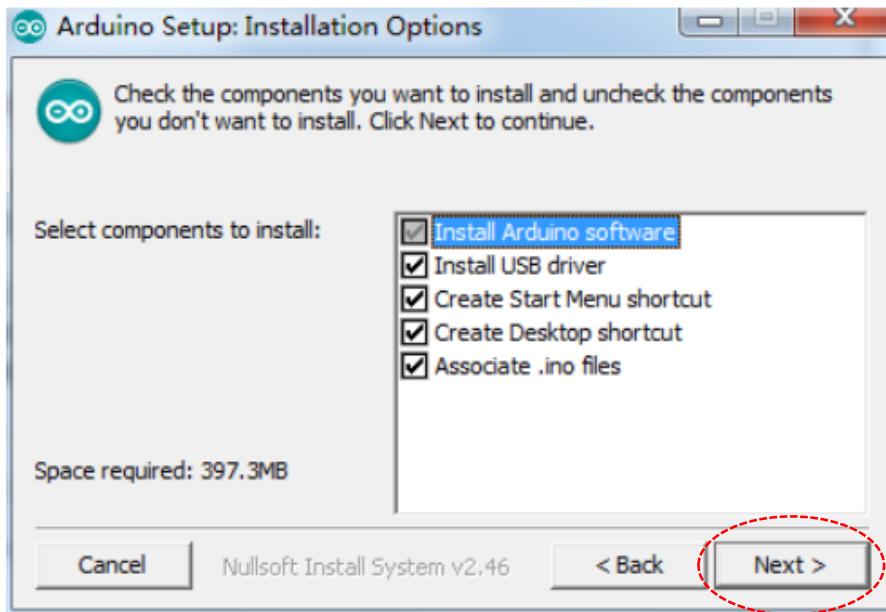
រូបភាព ១៖ គេហទំនាក់របស់ម្មាប់ដោនឡូតកម្មវិធីArduino

ការតាំងឡើងកម្មវិធី Arduino លើម្ចាស់ Window នៅពេលទាញយកកម្មវិធី Arduino IDE ត្រូវបើយធ្វើការដោយដឹងដូចខាងក្រោម៖

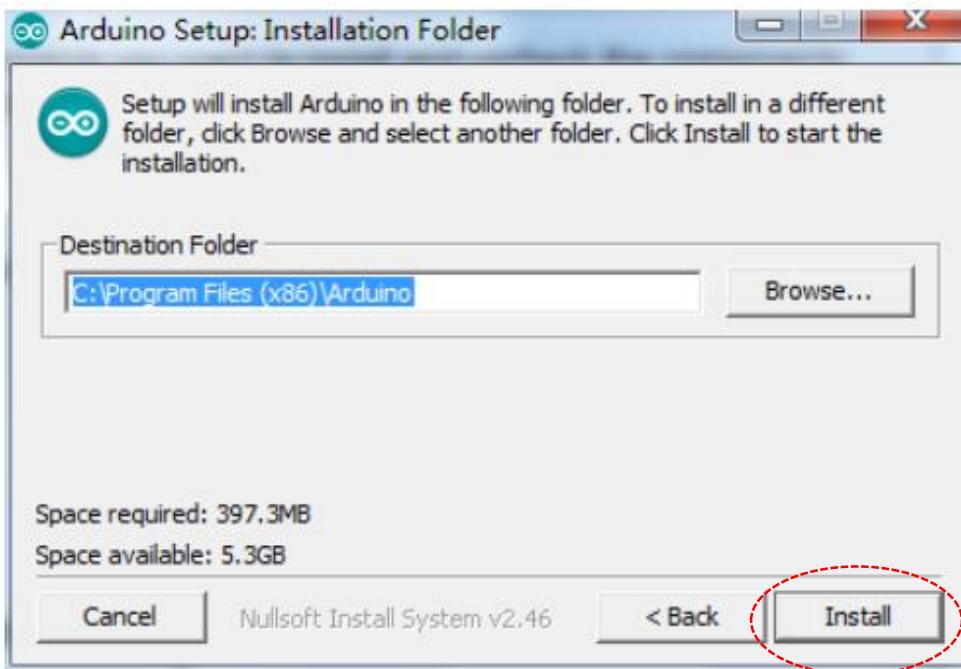
- ចូចយកពាក្យ I Agree នៅពេលយើត្រួចកម្មវិធីដូចខាងក្រោម៖



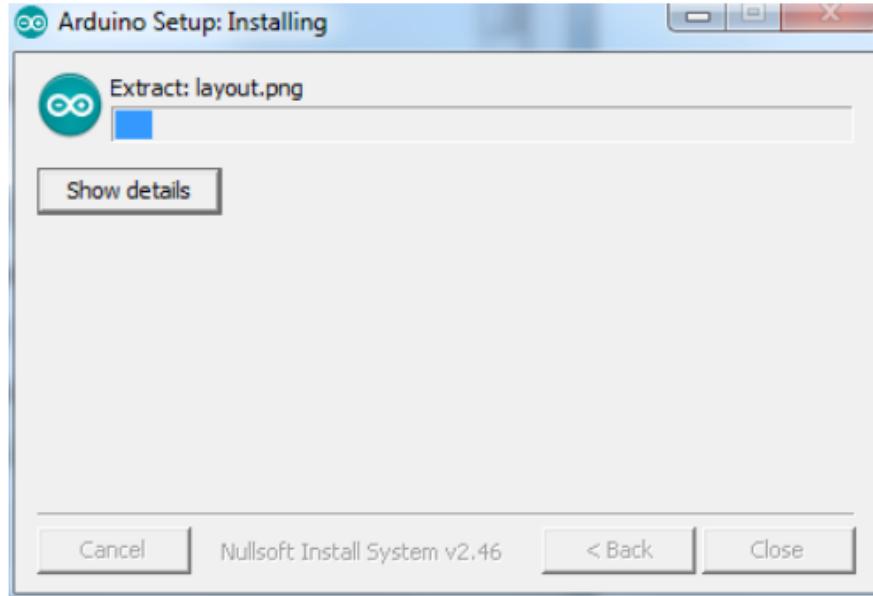
2. បន្ទាប់មកលោតផ្តាំងកម្មវិធីថែល្អមក នូចចុចពាក្យ Next



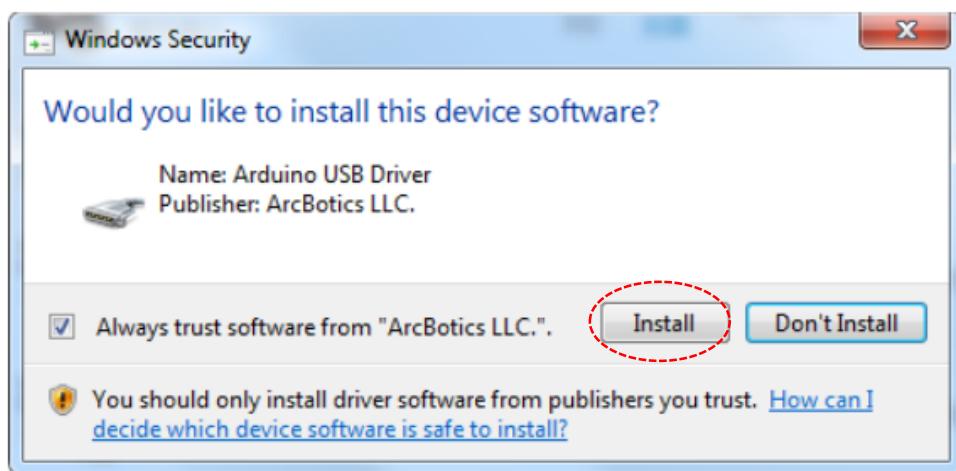
3. បន្ទាប់ពីចុចពាក្យ Next មកលោតផ្តាំងកម្មវិធីថែល្អមកទៅតែ បង្ហាញពីទីតាំងកម្មវិធីសម្រាប់ខ្លួន ការប្រើប្រាស់ការព្រឹងគឺជាដាច់ក ទន្លឹនយ នៅ drive C ករណីនេះអាចដាក់បាន ត្រាន់ពេចចុចពាក្យ Browse... បើមិនដូរទីតាំងកំណត់បានដែរ បន្ទាប់មកអាចតាំឡើងកម្មវិធីនេះបានត្រាន់ពេចចុច Install



4. បន្ទាប់ពីចូចពាក្យ Install វានីអេឡាត្រូចផ្តល់កម្មវិធីដូចខាងក្រោម ហើយនៅចាំរហូតដល់ចប់ នឹងលោកពាក្យ complete បន្ទាប់មកចូចពាក្យ close



5. ករណីនៃចាំរហូតដល់វាលោកពាក្យ Complete វានីអេឡាត្រូចផ្តល់កម្មវិធីមួយឡើត បើនេះផ្តល់កម្មវិធីនេះ កំពុងរដ្ឋខ្លួន និងកំពុងរដ្ឋអតិថជញ្ញាសាស្ត្រ នៅពេលដោឡូបានឡើង គាត់ចេញមួយដង កំពុងរដ្ឋខ្លួន គាត់ចេញតីរដង ហើយចេញត្រូវដ្ឋីសបីស Install របុត ហាមចូច ពាក្យ Don't Install ហើយកចូច Don't Install អាចកំពុងរបស់អ្នកមិនស្ថាល់ជាមួយគ្រឹះរបស់Arduino ទេ

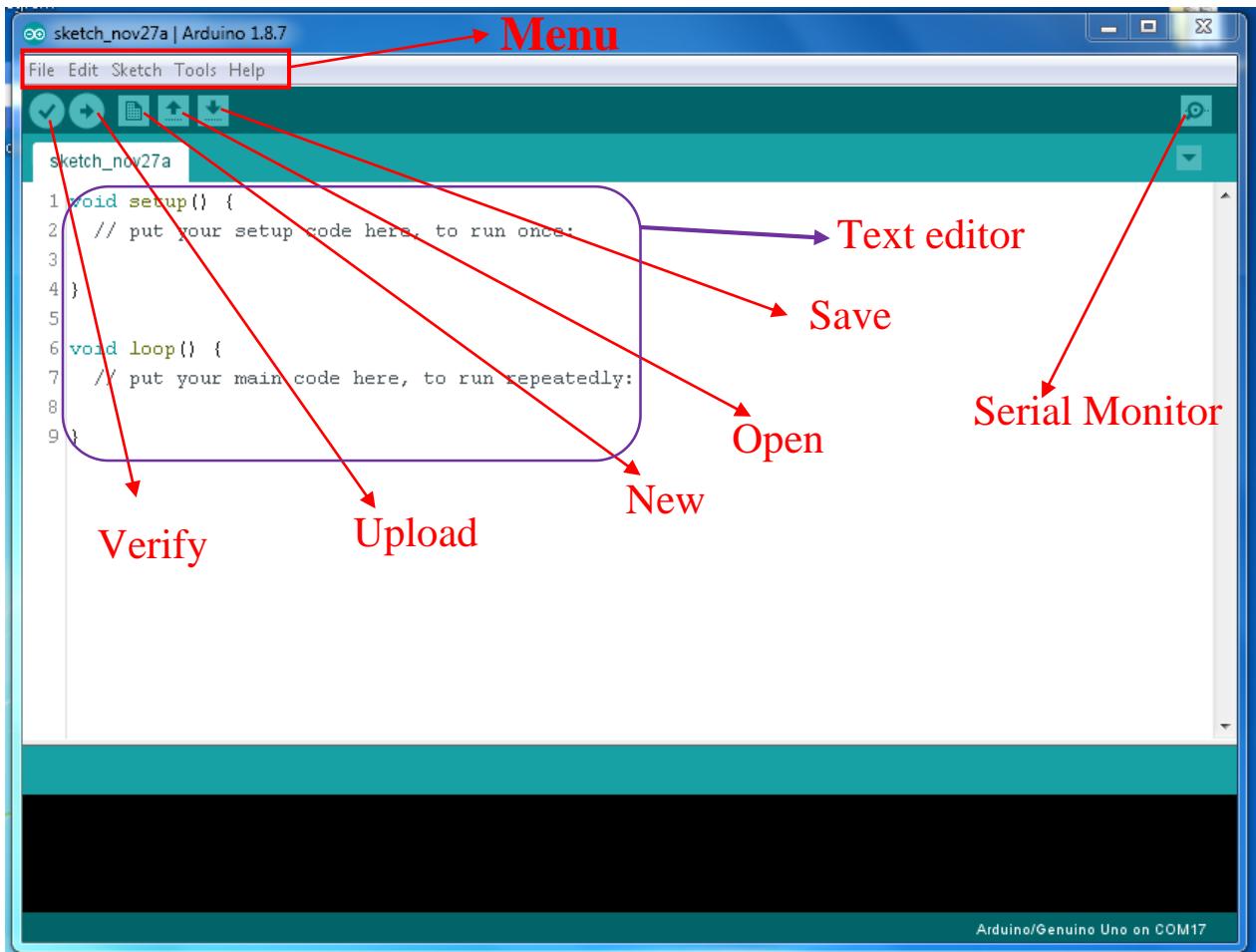


6. បន្ទាប់ពីចូចពាក្យ close វានីអេឡាត្រូចផ្តល់កម្មវិធី ហើយរាបឆ្លាស់ Icon នៅលើ desktop ដូចខាងក្រោម ៖



រូបភាព ២៖ រូបកម្មវិធី Arduino ដែលបង្ហាញនៅលើ Desktop កំពូទ័រ

២. ផ្តល់នូវជុំឡាយទិន្នន័យ Arduino



រូបភាព ៣៖ ផ្តល់នូវជុំឡាយទិន្នន័យ Arduino

Menu ៖ គ្មានកម្មវិធី Arduino មានប្រព័ន្ធឌី File , Edit , Sketch ,Tools និង Help ។ នៅពេលចួចធ្វើការនឹងយុទ្ធសាស្ត្រ នឹងបង្ហាញផ្លូវកញ្ចប់ជាប្រព័ន្ធនៅតួនាទី។

Verify ៖ ត្រូវបានបញ្ជីសម្រាប់ផ្តល់ជាតិក្នុងដែលសរសរចានាថុសវិនិត្ត។

Upload : ប្រើសម្រាប់បញ្ចូលកូដទៅក្នុង Arduino នៅពេលដែលសរស់រក្សាទុកតែមានខុស

New : ប្រើសម្រាប់បង្កើតផ្តាំងកូដថ្មី

Open : ប្រើសម្រាប់ទាញយកកូដដែលសរស់រក្សាទុកនៅត្រង់ណាមួយ

Save : ប្រើសម្រាប់រក្សាទុកកូដនៅត្រង់កន្លែងណាមួយនៅក្នុងទីតាំង

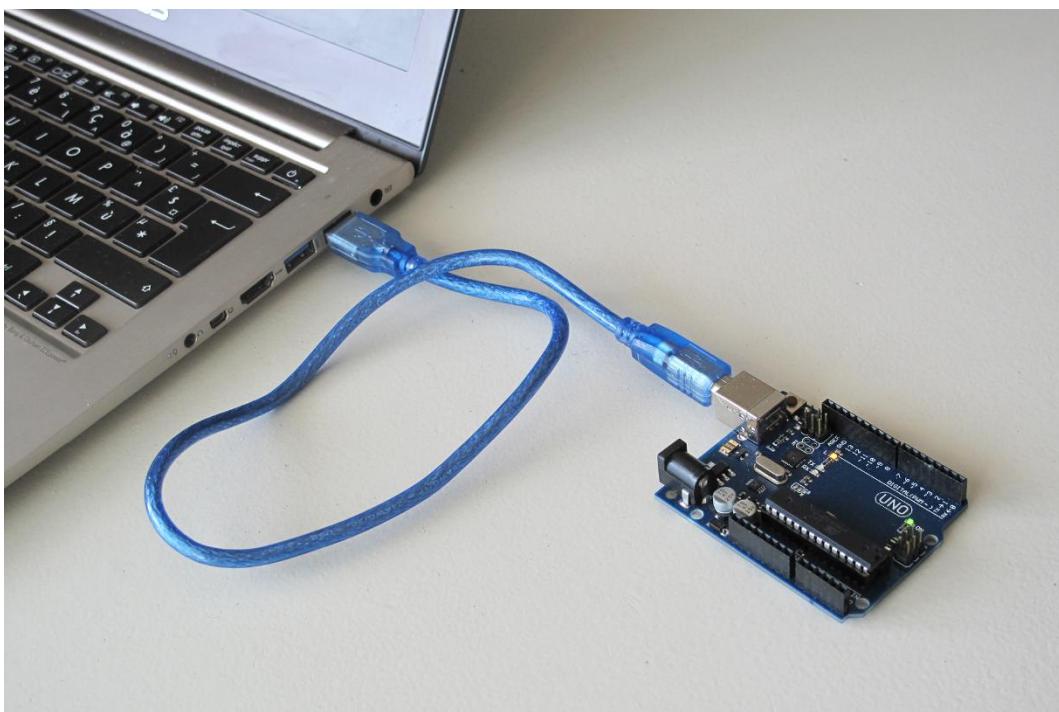
Serial Monitor : ប្រើសម្រាប់បញ្ជាផ្ទៃទូលទិន្នន័យរវាង Arduino និង កុំព្យូទ័រ

Text editor : គឺជាកំន្លែងសរស់រក្សាទុកនិងកូដហើយ ហើយអ្នកយើងចាក្យ void setup និង void loop ។
void setup () : កូដដែលសរស់រក្សាទុកនៅក្នុង void setup() : គឺមាននឹងយ៉ាងរាយរាយដំណើរការហើយ
void loop () : កូដដែលសរស់រក្សាទុកនៅក្នុង void loop () គឺសម្រាប់សរស់រក្សាទុកដំណើរការហើយ

តាមលក្ខណនាប់អ្នកសរស់រក្សាទុកនៅក្នុង void loop () គឺមាននឹងយ៉ាងរាយរាយដំណើរការហើយ

៣. រច្ឆេទនោះ នឹង Arduino ចាប់ឡើងទីនេះ

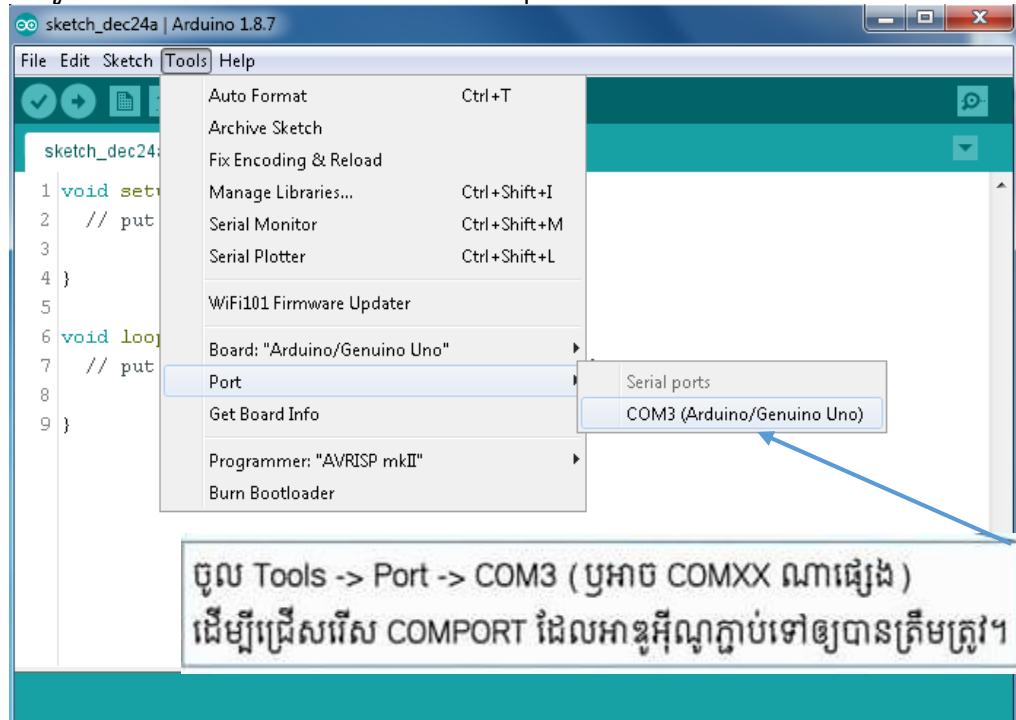
ដើម្បីតាមតាមរបស់អ្នកសរស់រក្សាទុក ទៅនឹងកុំព្យូទ័រអ្នកត្រូវប្រើប្រាស់ USB ដូចខាងក្រោមទី៤ ។



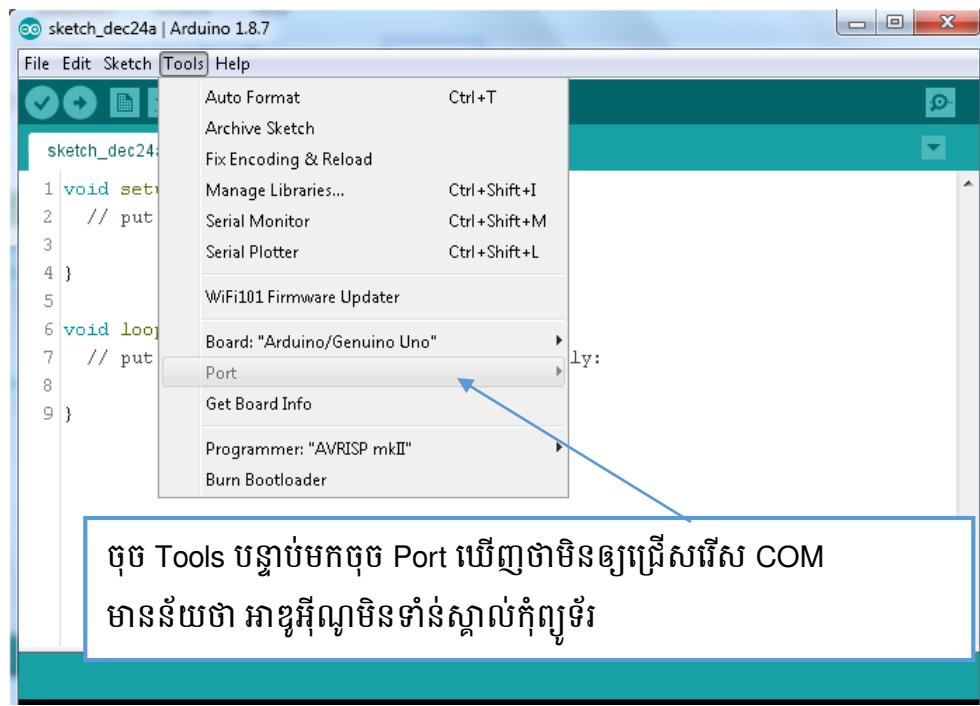
រូបភាព ៤: ផ្តាំងអាមុន្តុល្អបន្ថែមទៅនឹងកុំព្យូទ័រ

នៅពេលអ្នកតាមតាមរបស់អាមុន្តុល្អទៅនឹងកុំព្យូទ័ររួចហើយចង់ដើងថាកុំព្យូទ័រស្ថាល់អាមុន្តុល្អប្រអត់
 ត្រាន់តែចូលលើផ្តាំង Menu ហើយចូចលើពាក្យ Tools ហើយចូចលើពាក្យ Port បន្ទាប់មកទៀតដើរ

នឹង COM ដែលបញ្ជាក់នៅខាងក្រោម COM មានពេញ Arduino ហើយបើចុចការណ៍ត្រួតពីខាងក្រោម។



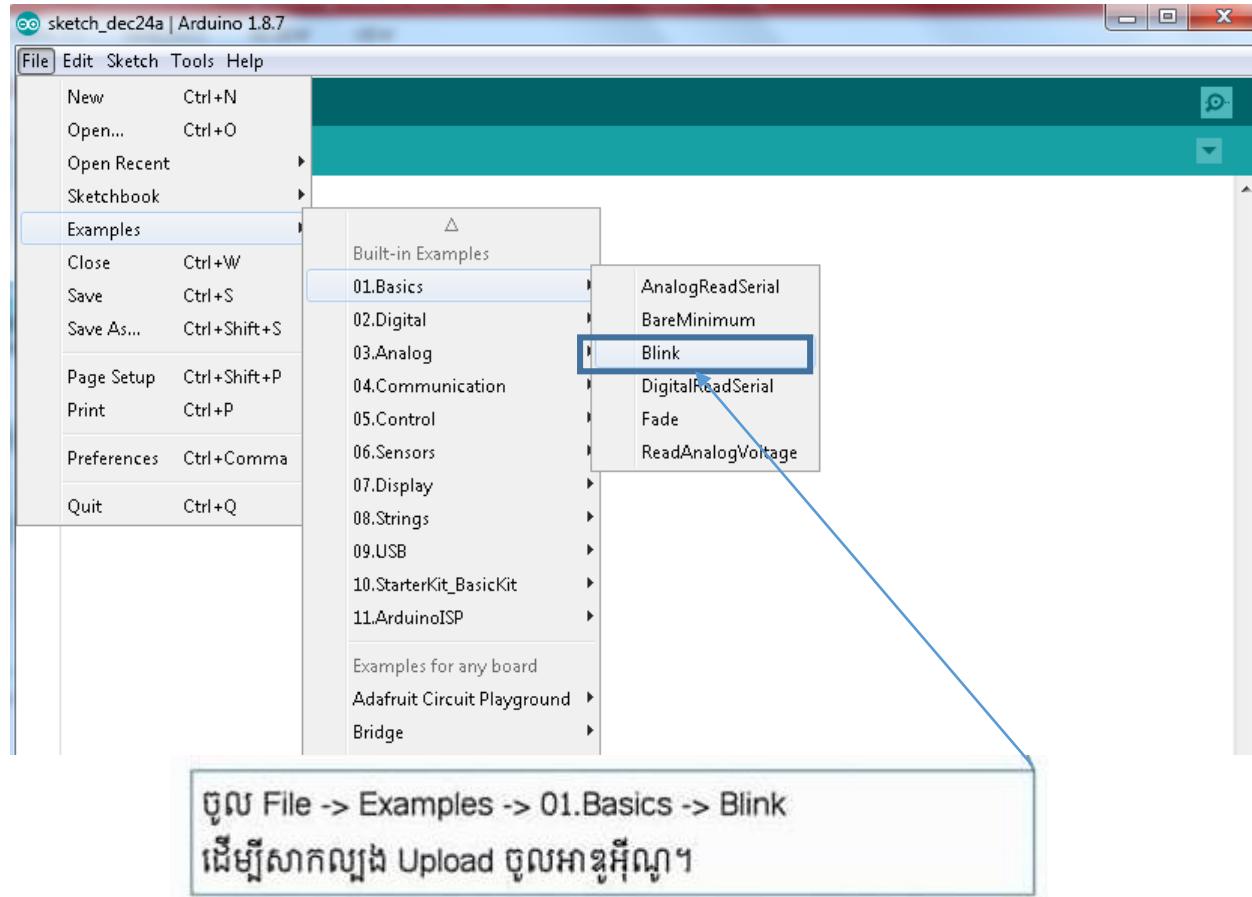
រូបភាព ៥: ការជួនអាមុគ្គលាស្ថាល់ជាមួយកំពុង



រូបភាព ៦: ការជួនអាមុគ្គលាអិនស្ថាល់ជាមួយកំពុង

៤. ការតែនលេខនៃការអនុវត្តន៍នៃ Arduino ជាមួយកុំព្យូទ័រ

ការពេសសាកល្បងធ្វើនៃ Arduino ជាមួយកុំព្យូទ័រដូចខាងក្រោមតាមលំដាប់លំដោយ
ត្រូវប្រាកដថាកុំព្យូទ័របស់អ្នកបានត្រូវបានធ្វើនៃ Arduino Uno ជាមួយខ្សោយ USB និងភាគតាមលំដាប់លំដោយ
ត្រូវប្រាកដថាកុំព្យូទ័របស់អ្នកបានធ្វើនៃ Arduino Uno ជាមួយខ្សោយ USB និងភាគតាមលំដាប់លំដោយ



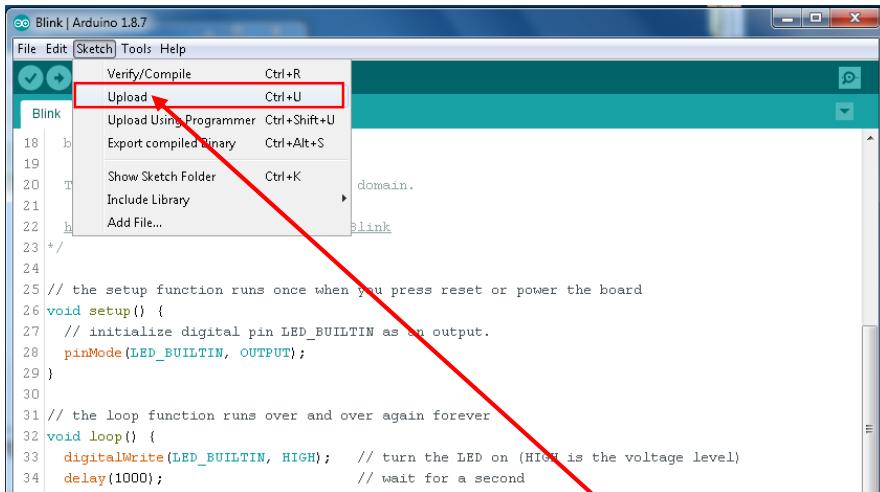
រូបភាព ៧៖ ការជើរពេសសាកល្បងឡើងមុនពីកម្រិតបិទបីក(Blink) របៀបចូល

```

Blink | Arduino 1.8.7
File Edit Sketch Tools Help
Blink
18 by Colby Newman
19
20 This example code is in the public domain.
21
22 http://www.arduino.cc/en/Tutorial/Blink
23 /*
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
34   delay(1000);                      // wait for a second
35   digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36   delay(1000);                      // wait for a second
37 }

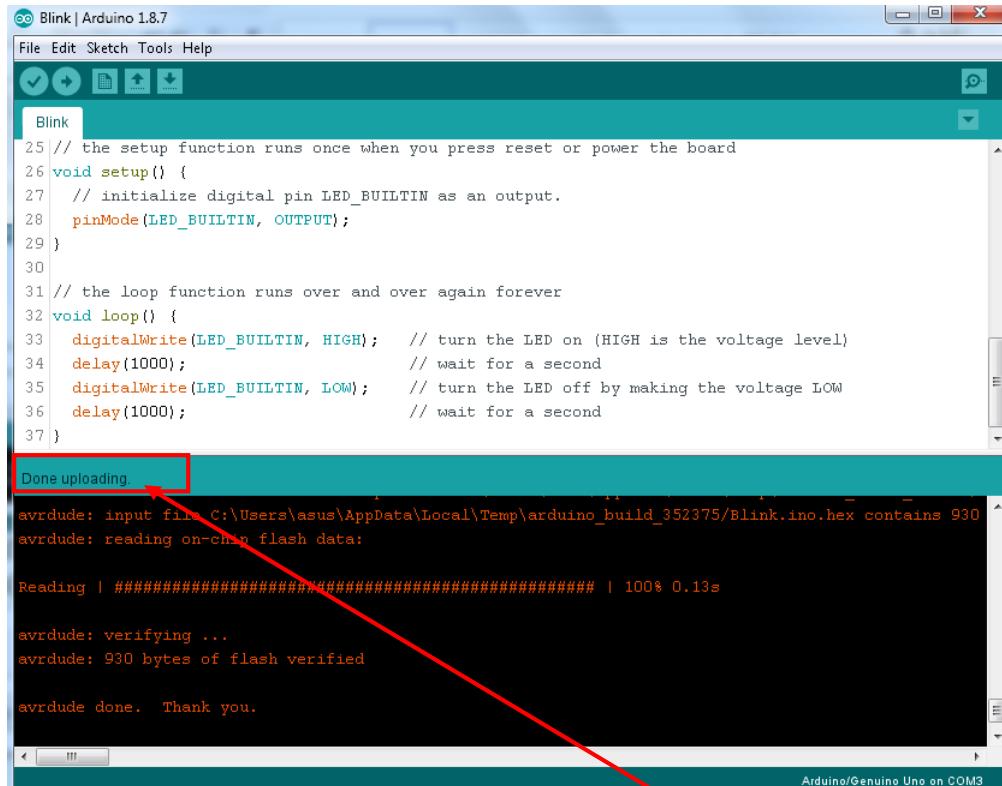
```

រូបភាព ៩: ក្នុងដែលមានសម្រាប់តែសាកល្បងចូអំពូលបីក(Blink)



**ចូល Sketch -> Upload ដើម្បី Compile និងអាប់ខ្សែត (បញ្ជូន)
ទៅក្នុងអាមីនីណុយ។**

រូបភាព ១០: រៀបអាប់ខ្សែតក្នុងជាក់ទៅក្នុងអាមីនីណុយ



```

Blink | Arduino 1.8.7
File Edit Sketch Tools Help
Blink
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }

Done uploading.

avrduude: input file C:\Users\asus\AppData\Local\Temp\arduino_build_352375/Blink.ino.hex contains 930
avrduude: reading on-chip flash data:

Reading | ##### | 100% 0.13s

avrduude: verifying ...
avrduude: 930 bytes of flash verified

avrduude done. Thank you.

Arduino/Genuino Uno on COM3

```

ពេលអាប់ទ្វាគ្រប់ រានីដែលសារដូចខាងលើ ។
បន្ទាប់មកសង្គតមិនអាចអីណាមុនបែសមូក កំពុងវិភីត្រឹងបិទ/បើក ។

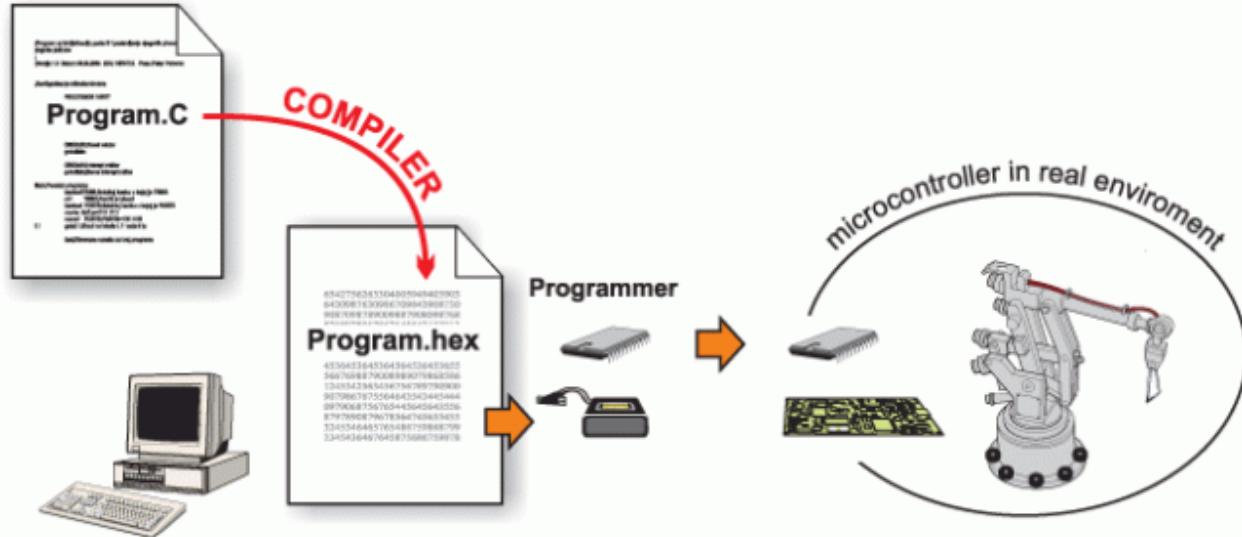
រូបភាព ១០៖ កន្លែងបង្ហាញថាក្នុងពាណិជ្ជកម្មភូលក្នុងអាម៉ូដ្ឋូចហើយ



រូបភាព ១១៖ អាម៉ូដ្ឋូនឹងភ្លើលតែភ្លើលតែដែរនៅពេលអាប់ទ្វាគ្រប់

ចំណែក ៤ ៖ ផែលវាំគម្រោង និងរបៀបអនុវត្តន៍

មិនត្រូវដឹងទៀត តារ់ផែលវាំគម្រោងនៃភាសាកម្រោង C



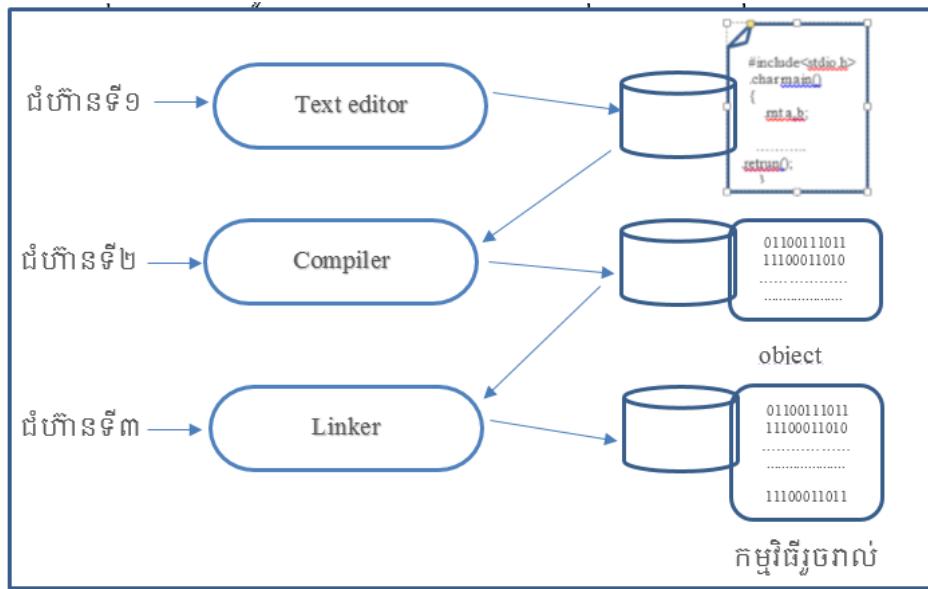
រូបភាព ១ ៖ ផ្គារក្រាមកូដទៅលើ Microcontroller

១. ប្រព័ន្ធឌីតាលិកសាឈា C

ភាសា C គឺជាការសាកម្បិធីមានរចនាសម្ព័ន្ធមួយ ដែលត្រូវបានគេប្រើប្រាស់ដើម្បីសរស់រក្សាទុកដាក់ សម្រាប់ដោះស្រាយបញ្ហាញ្តាល់ដែលបាន ដោយមិនចាំបាច់ព្រមទាំងមិនមែនការសាកម្បិធីទេ ក្នុងការសរស់រក្សាទុកដោះស្រាយ ។ នៅក្នុងឆ្នាំ 1967 លោក Martin Richards បានស្ថាបនាអេង្សីនុវកសាស្ត្រ ដាក់លេខាជាមី ឬ Basic Combine Programming Language ហើរកាត់ថា BCPL ។ ក្រោយមកទៀត ក្នុងឆ្នាំ 1970 លោក Ken Thompson បានបន្ថុស្ថាដែនេះដោយបង្កើតបានការបានលក្ខណៈ ស្របដៃនីត្តាដោយដាក់លេខាជាមី B ។ នៅទីបញ្ហាប់ក្នុងឆ្នាំ 1972 លោក Dennis Ritchie បានស្ថាបនាការសាកម្បិធីមួយទៀត ថា C ។

២. កម្រោង C នៃក្នុងភាសា C

ម៉ាសីនកំពុទ្ធអម្ចាយយល់កម្បិធីមួយបាន ឬបោះព្រែកកម្បិធីទោះត្រូវបានបំលែងទៅក្នុងភាសាម៉ាសីន ។ សម្រាប់កម្បិធីដែលបានបែរទេរនឹងក្នុងភាសា C ត្រូវបានដោយ ទំនួរបន្ទាត់ ឬ ក្រោមឯកសារ ឬ និងអ្នកគេយល់ដឹងដឹងដែរពីជំហានទាំងនេះ ។ ត្រូវបានគើតឡើងជាប្រើប្រាស់ដើម្បី នៅក្នុងកម្បិធីការកែកកំហុសនិងធ្វើការកែកលំអក្សុងម៉ោង លក្ខណៈកាន់តែប្រសើរឡើង ។ កម្បិធីមួយដែលអាចវិនិច្ឆ័យឡើងនៅក្នុងភាសា C ត្រូវអនុវត្តតាម ជំហានទាំងបីដូចក្នុងបង្ហាញនៅក្នុងរូបខាងក្រោម៖



រូបភាព ២ : ដំហែនទាំង៣ ក្នុងការអភិវឌ្ឍន៍កម្មវិធីក្នុងភាសា C

២.១. ភាសានេះ និងការតាមដានកម្មវិធី (បំបាត់ខ្លួន)

កម្មវិធីប្រពេទជាពាណិជ្ជកម្មយុទ្ធសាស្ត្រ និងជូន ទិន្នន័យជាតិ ទាំងឡាយរបស់កម្មវិធីភាសាដែលមាននៅក្នុងម៉ាសីនគោចសរស់បុគ្គលិក របាយការណ៍និងសរស់របស់កម្មវិធីទាំងឡាយ។ ភាពខុសប្រព័ន្ធ ត្រូវរាយការប្រព័ន្ធបន្ទាត់ទាំងឡាយនៃក្នុងខណៈដែលការប្រព័ន្ធបន្ទាត់ទាំងឡាយត្រូវបានសរស់ឡើង ដោយការប្រើប្រាស់បញ្ហាត់ទាំងឡាយនៃក្នុងខណៈដែលការប្រព័ន្ធបន្ទាត់ទាំងឡាយនិងអនុបន្ទាត់សម្រាប់បញ្ហាត់ទាំងឡាយ។ បញ្ហាត់កម្មវិធីមួយត្រូវបានគេសរស់រចប់ វាត្រូវបានត្រូវក្រឡាតក ជាងកសារនៅក្នុងខិស់ ។ ឯកសារនេះត្រូវបានគេហេត្តិជាងកសារប្រភពដែលត្រូវបានគេបញ្ចូនដើម្បីជាឌីជី Input របស់ Compiler ។

២.២. ភាព Complile ឥឡូវ (បំបាត់ខ្លួន)

ក្នុងដែលមាននៅក្នុងឯកសារដែលមាននៅក្នុងខិស់ ត្រូវបានបកប្រើប្រាស់ជាផាក្យម៉ាសីន ហើយ ការដោរនេះត្រូវបានសម្រេចឡើងដោយ Compiler ។ Compiler របស់ភាសាកម្មវិធី C ត្រូវបានអនុវត្តឡើងដោយកម្មវិធីពីជាត់ចេញពីត្រូវ ឬ Preprocessor និង Translator ។ ការដោរ របស់processor នានា ឯកសារប្រភពនិងរៀបចំថាត់ចេងឡើងសម្រាប់បញ្ចូនទៅឲ្យ translator ។ នៅក្នុងខណៈដែលការមើលបញ្ជាតិសេសទាំងឡាយ ដែលវាស្ថាល់ចាត់ជាង Preprocessor directive ។ Directive ទាំងឡាយបានបញ្ជាតិសេសទាំងឡាយដើម្បីធ្វើការជាក់ជំនួស ឬពិនិត្យមើល libraries ក្នុងពិសេសទាំងឡាយដើម្បីធ្វើការជាក់ជំនួស

នៅក្នុង ក្នុង និងរៀបចំក្នុងសម្រាប់បកប្រែទៅជាកាសាម៉ាសីន ។ លទ្ធផលនៃការរៀបចំមុនប្រពិបត្តិត្រូវបានគេហោចាតាដោយ translator ក្នុងអនុវត្តការងារពិភាក្សាលើការបំលែងកម្មវិធី ឲ្យទៅជាកាសាម៉ាសីន ។

Translator អាន translation unit និងសរស់បានជា object module (ក្នុងនៅក្នុងភាសា ម៉ាសីន) ទៅកាន់ឯកសារមួយហើយបញ្ជាត់មកត្រូវបានគេធ្វើដោយនឹងធ្វើឡើងដែលបាន Compile ទុកមក ពីមុនដោយឡើងទៀតដើម្បីបង្កើតបានជាកម្មវិធីពិតប្រាកដមួយឡើង។ ទៅ:ជាលទ្ធផលរបស់Compiler គឺជាក្នុងរបស់ម៉ាសីន យ៉ាងណាក់ដោយកំណែនិមិនទាន់ភាពប្រព័ន្ធបច្ចុប្បន្នបាននៅឡើយព្រោះ ថាការិមិនទាន់ មាន រូបច្បាសនូវ Function ដោយឡើងទៀតដើម្បីការចំណាំរបស់ ភាសាកម្មវិធី C ។

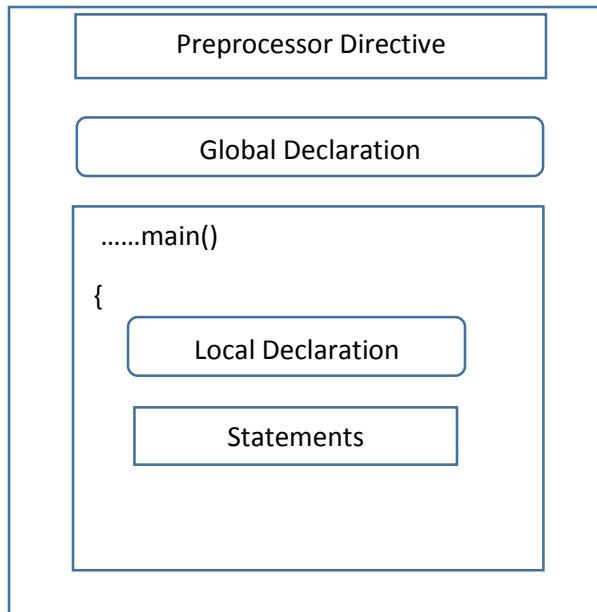
២.៣. គារគ្រប់គម្រោង (បំបាត់បាន)

iii. ຂຶ້ນຫຼືເຕະຫົວເຫັນສົ່ງຂອງທີ່ໃຫ້ຂູ້ລັກສຳ C

ផ្សែងរាល់កម្មវិធីក្នុងភាសា C ដូចឡើងពីដោយប្រកាសសកលម្អួយនិងអនុគមន៍ជាប្រើនទៀតដោយប្រកាសសកលកំពើឡើងនៅចំនួចថាប់ដើមនៃកម្មវិធី។ មានពេលអនុគមន៍តែម្អួយប៉ុណ្ណោះនៅក្នុងកម្មវិធីត្រូវបានគេដាក់ឈ្មោះថាគាត់ main() ដែលជាចំនួចថាប់ដើមនៃកម្មវិធី។ នៅក្នុងកម្មវិធីត្រូវបែងអនុគមន៍ទាំងអស់របស់ខ្លួនដោយប្រាកាសគេហទ័របានគេបែងចែកថ្លាបាតីរៀងក្រកីដ្ឋានឯមនៃយុទ្ធសាស្ត្រ និងដោយការងារអនុវត្ត។ ដោយក្រកីយមនៃយុទ្ធសាស្ត្រ គឺជាចំនួចថាប់ដើមនៃអនុគមន៍ដោយមានអធិប្រាយពីទិន្នន័យដែលនិងត្រូវបានគេយកមកប្រើប្រាស់ក្នុងអនុគមន៍។

ចំពោះអនុគមន៍ ផ្តើកនេះត្រូវបានគេស្ថាល់ថាគាតិយមនៃយោងក្នុង ។ ផ្តើកការងារអនុវត្តតាមបន្ទាប់ពី ផ្តើកនិយមនៃយោងដោយក្នុងនោះមានផ្តុកបណ្តាលសេចក្តីណែនាំចំពោះ កំពុទ្ធដែលមែបប្រើបាស់ឡើងអនុវត្តនិភី ដូចជា ការបុកពីចំនួនជាអើម។ នៅក្នុងភាសា C សេចក្តីណែនាំទាំងនេះត្រូវបានសរសេរឡើង ព្យាកាមទាំងនេះនៅការងារអនុវត្តដែលដឹងលើលទ្ធផលទូរយើងនៅផ្តើកជួចចាប់។ រឿងភាពចាប់ជាបារិយាណាព្យាបាលប្រចាំថ្ងៃ

នៃក្នុងក្នុងកម្មវិធីផ្ទុកមួយសរសរឡើងនៅក្នុង ភាសា C ។ នៅក្នុងខាងក្រោមនេះកម្មវិធីមួយត្រូវបានផ្តល់ឡើងពីអនុគមន៍ main() តែមួយគត់ Preprocessor directive ដីមានសារ៖ សំខាន់មួយដែលប្រើប្រាស់នៅក្នុងគ្រប់កម្មវិធីតិចា include ហើយប្រាប់ថា កម្មវិធីមានរូបរាងនៅក្នុងខាងក្រោមនេះជាលើ library ជាអាជាហរណ៍ #include <stdio.h>)



រូបភាព ៣ ៖ ទំនើបបច្ចន្ទរបស់កម្មវិធីក្នុងភាសា C

```

#include <stdio.h>
Void main()
{
    Printf(" welcome to RPITSSR");
}
  
```

រូបភាព ៤ ៖ ឧទាហរណ៍របៀបង្ហាញសរសរកម្មវិធីលើកដំបូង

លេខខ្លួនក្នុងក្រុមការណ៍

១. ជាតុច្បាស់ដែលគឺជាកម្មវិធី

រាល់ការសរស់រក្សាទិន្នន័យដោយបង្កើតឡើងដោយធានាតុសំខាន់បូន្មុមជាងក្រោម៖

- ❖ **Expression** : កន្លែមបង្កើតឡើងដោយ operands និង operator ។ 1 operand ជាធិន្នន័យដែល operator ត្រូវការ ។ 1 operator គឺជាប្រមាណវិធីគិតវិញ្ញាបុគ្នាសុំកន្លែកឡើង ។

ឧទាហរណ៍ : $X - Y$

$A + B$

$A - 200$

- ❖ **Statements** : គឺជាពាក្យបញ្ជាផ្លាមួយ ។ រាល់ Statements ត្រូវបញ្ចប់ដោយ semicolon (;) ។

ឧទាហរណ៍ : $a = 50 ;$

$A = b+C ;$

$Sum = Total +A ;$

- ❖ **Statement Blocks** : បណ្តាំនៃ Statements ជាប្រើន ។ ប៉ុណ្ណោះ Compiler ចាត់ទុកចាប់ពី នៃ Statements ទាំងនេះបង្កើតឡើងជាតុ statement ដើម្បីកំណត់បញ្ជី ។ ចូរពិនិត្យមើលឧទាហរណ៍ខាងក្រោម សញ្ញា “ $>=$ ” មានន័យថាចាំជាង បុស្សី ។

ឧទាហរណ៍ : if (motor_speed $>= 100$)

{

```
    slow_down ();
    read_motor_speed();
```

}

- ❖ **Function Blocks** : គឺជាបណ្តាំនៃកូដដែលបង្កើតឡើងដើម្បីធ្វើការកំណត់បញ្ជី ។ ក្នុងឧទាហរណ៍ខាងលើយើងមាន function ពីរគឺ slow_down() និង read_motor_speed() ដែលពួកវាត្រូវបានសរស់រក្សានៅក្នុងគោលបំណងសម្រាប់កូដការកំណត់បញ្ជី ។

២. តារាងធិន្ទយោ (Comment)

Comment គឺជាមួយប្រយោត្រប្រើសម្រាប់អធិប្បាយអត្ថន័យរបស់ក្នុង បូប្រើសម្រាប់អ្នកសរស់រក្សាទីដើម្បីអាយងាយស្រួលក្នុងការអាជីវកម្ម។ Compiler មិនបានរួមបញ្ចូលការអធិប្បាយនៅក្នុងការបញ្ចូនក្នុង និងបំលែងទៅជាកម្មវិធីទេ។

ក្នុងការជើរការដោយ Comment មានពីរប៉ែបគឺ single-line comments និង multi-line comments។

- ❖ **single-line comments** បាប់ធើមដោយសញ្ញា (//) ឬ សញ្ញា (//) អាចសរស់ Comments បានតែមួយបន្ទាត់តែប៉ុណ្ណោះ ។
ឧទាហរណ៍ : // LED is connect to pin 5
 // push button b1 led2 is no
 // delay 500 milliseconds
- ❖ **multi-line comments** បាប់ធើមដោយសញ្ញា (/*) និងបញ្ចប់ទៅវិញដោយសញ្ញា (*/) ។ អាចសរស់ Comments បានចិនបន្ទាត់តែត្រូវ ធើមដោយសញ្ញា (/*) និងបញ្ចប់ទៅវិញដោយសញ្ញា (*/) ។
ឧទាហរណ៍ : /* LED is connect to pin 5
 push button b1 LED is NO
 delay 500 milliseconds */

៣. ទម្រង់នៃការបញ្ចូន C នូវកម្មវិធី

នៅពេលដែលអ្នកបាប់បើកកម្មវិធី Arduino ត្រាមនោះអ្នកនឹងបាប់អារម្មណ៍យើងៗ មានតំបន់ពីរដែលខ្ពស់ពីគ្មានចូលរួមបានក្រោម ដែលជាកន្លែងសម្រាប់អ្នកសរស់រក្សាទី។

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

}

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

}
```

- **void setup ()** : គឺជាការរៀបចំ ។ ការរៀបចំមានដូចជាការប្រកាសអចេរដើម្បីប្រើប្រាស់ក្នុងកម្មវិធីការកំណត់ទិសដោនៃដើម្បីចូល ប្រការកំណត់ប្រភេទនៃលំហូរ ទិន្នន័យជា អាណាព្យាក ប្រើឱដីចលជាដើម ។ជាយុទ្ធសាស្ត្រ void setup () គឺមាននំយចា កំណត់ដើម្បីរបស់ ការអ្នកអើណុ រាយការដំណើរការ ហើយមាន ក្រុមក្រាម (Syntax) សម្រាប់សរស់របស់ ។
- ឧទាហរណ៍ :**

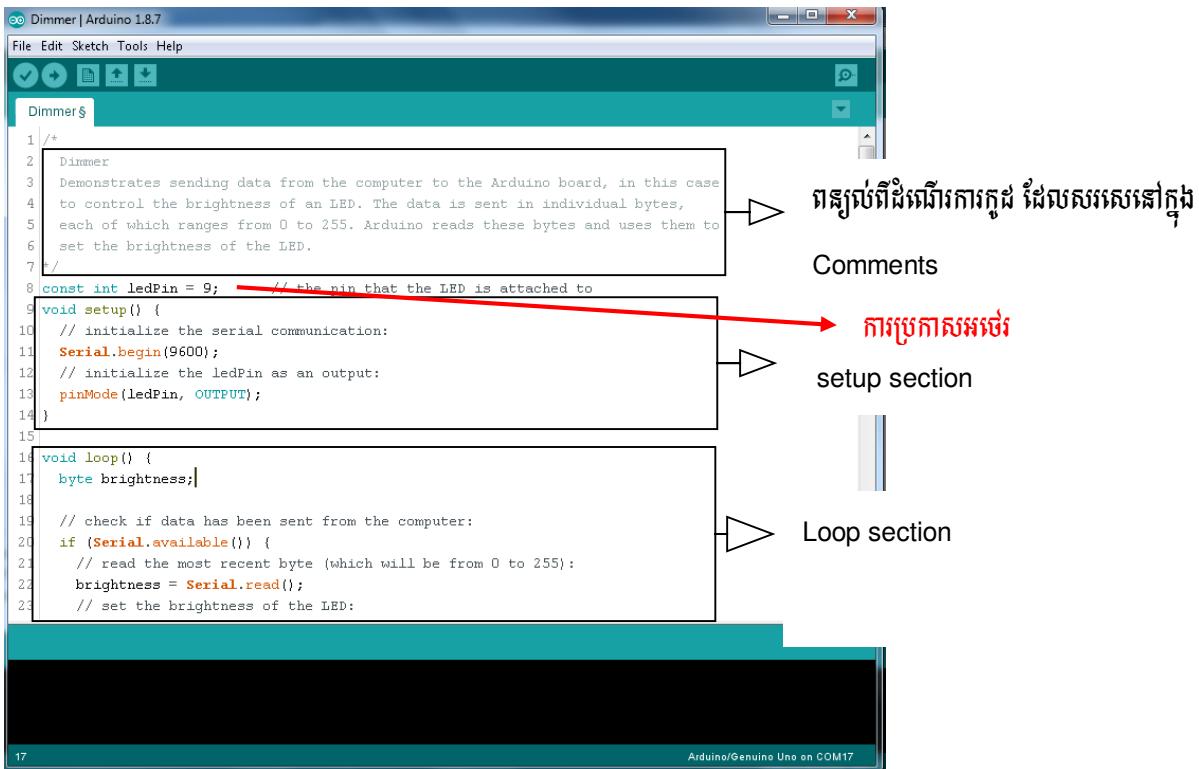
```
void setup ()  
{  
    pinMode ( pin , OUTPUT ) ; // sets the pin as output  
}
```

- **void loop ()** : គឺជាប្រព័ន្ធបន្ទីរសារចុះសារទ្វីននៃក្នុង ដូចជាអាជីវិតទិន្នន័យចូលពីដើម្បីរបស់អ្នកអើណុ រាយ ប្រការបញ្ជាផ្ទាល់ទិន្នន័យតាមដើម្បីរបស់អ្នកអើណុ រាយ ហើយសម្រាប់សរស់រដំណើរការអ្នកអើណុ រាយ តាមលក្ខខណ្ឌរបស់អ្នកអើណុ រាយ ។

ឧទាហរណ៍ :

```
void loop ()  
{  
    digitalWrite ( pin , HIGH ) ; // sets " pin " to ON  
    delay ( 1000 ) ; // pauses for 1 seconds  
    digitalWrite ( pin , LOW ) ; // sets " pin " to OFF  
    delay ( 1000 ) ; // pauses for 1 seconds again  
}
```

នៅក្នុងការអនុវត្តន៍ជាក់ស្អែង ប្រើឱលក្ខដនោលើអ្នកអើណុដើម្បីរួចរាល់ រាយការប្រកាសអចេរ (variable declaration)ជាដើម ។ សញ្ញា { } ហើយជាឍូកសម្រាប់ ផ្តល់បណ្តុះនៃក្នុងឯកសារយើងដូចម្នាយ ដូចជាថាបន្ទីរសារចុះសារទ្វីននៃក្នុងការប្រកាសអចេរ (variable declaration)ជាដើម ។



៤. ប្រភេទទិន្នន័យ (Data type)

ប្រភេទទិន្នន័យភាសា C គឺជាអ្នកឈ្មោះគោរពថាគាមប្រភេទទិន្នន័យស្ថីជា ANSI C ប៉ុន្តែ មានករណីលើកលែងមួយចំនួន ។

អចិវ (variable) គឺជាទីតាំងក្នុង memory សម្រាប់ក្រាមទិន្នន័យ ។ ពេលយើងសរស់រក្សាទី យើងត្រូវកំណត់ប្រភេទទិន្នន័យ (data type) ឲ្យអចិវដែលយើងប្រើប្រាស់ ។ ការកំណត់ប្រភេទទិន្នន័យ ឲ្យអចិវមានសារ៖ សំខាន់ ពីរបោះរាងកំណត់ទិន្នន័យទាំងអស់ និងការប្រើប្រាស់ទិន្នន័យ ។

តារាង ១ : ប្រភេទទិន្នន័យក្នុងភាសា C

ប្រភេទទិន្នន័យក្នុងភាសា C		
ប្រភេទ	ចំនួន Byte	ដែនកំណត់
Boolean	1	អាចជាក្នុងសីកពិត (True) ឬ មិនពិត (False)
Char	1	-128 ទៅ 127
unsigned char	1	0 ដល់ 255

Byte	1	0 ដល់ 255
Int	2	-32768 ដល់ 32767
unsigned int	2	0 ដល់ 65535
Word	2	0 ដល់ 65535
Long	4	-2 147 483 648 ដល់ 2 147 483 647
unsigned long	4	0 ដល់ 4 294 967 295
Float	4	-3.4028235E+38 ដល់ 3.4028235E+38
double	4	-3.4028235E+38 ដល់ 3.4028235E+38
string	?	បណ្តុះនៃពួរអក្សរ
array	?	បណ្តុះនៃព័ម្ពលេខ
Void	0	និមិត្តសញ្ញាឯក្រប់សម្ងាត់ Function ដែលមិនបានតម្លៃគ្រប់មកវិញ ។

៥ . តារាងឈ្មោះអថេរ (Variable Name)

អថេរ (Variable) គឺជាមួយដែលត្រូវបានប្រើសម្រាប់កំណត់តំបន់នៃ Memory(RAM) ដែលត្រូវផ្តល់ទូទៅនៃ Data មិនទៀតទាំងអាស៊យទៅតាមការផ្តល់ព័ម្ពទៅអេយោរ។ ហើយឈ្មោះរបស់អថេរត្រូវបានកំណត់ដូចខាងក្រោម៖

- ឈ្មោះរបស់អថេរមួយត្រូវបានផ្តល់ព័ម្ពអក្សរ A-Z ឬ a-z ឬលេខ 0-9 អាចជាកំនៅចន្ទោះត្រូវបានប្រើបានដែលត្រូវបានដែលបង្ហាញសំនួរឈ្មោះរបស់អថេរ
- គេអាចប្រើ Underscore (_) ដីនូសការដែកយ្យា (Space Bar)
- ឈ្មោះរបស់អថេរមិនអាចដូចជាកើតិវត្សិភាព (Keyword) បានទេ
- មិនអនុញ្ញាតិអេយោរប្រើ Arithmetic Signs (+, -, *, /, %)
- មិនអនុញ្ញាតិអេយោរមានត្រូវបានសែនសង្គមជាតិ (. ` " @ # ! \$ & ...)

តារាង ២ : Keywords

Keywords				
absolute	data	if	return	typedef
Asm	defaault	inline	rx	typeid
At	delete	int	sft	typename
auto	do	io	short	union
Bit	double	long	signed	unsigned
Bool	else	mutable	sizeof	using
break	enum	namespace	static	virtual
case	explicit	operator	struct	void
catch	extern	org	switch	volatile
char	false	pascal	template	while
class	float	private	this	Loop
code	for	protected	throw	
const	friend	public	true	
Continue	goto	register	Try	

ឧត្តមាឌ់ : ampere , volt , _design , Day1

ឧងគ្រាមនេះគឺមិនត្រឹមត្រូវ ហើយត្រូវបានហាមយាត់ដោយ Compiler ខ្លះ

3day , ^home , %percent , this&that , not-good , what?.....

៤. ការប្រភាគនៃការបញ្ជី

អចេរមួយមុននឹងអាចប្រើប្រាស់បានគឺត្រូវប្រកាសជាមុនសិន។ ការប្រកាសអចេរមានន័យថា ប្រាប់ Compiler ត្រូវបានផ្តល់ព័ត៌មានថា អាសន្នរបស់អចេរមួយ នេះដើម្បីគោរចាយកមកប្រើក្នុង memory សម្រាប់ផ្តើកតម្លៃបណ្តាញ នៅពេលកម្ពុវិធីដំណើរការ។

នៅក្នុងការបញ្ជី

ប្រភេទទិន្នន័យ លើការបញ្ជី ;

Datatype Variable_name;

ឧចាងន៍ ៖

```

int sum ; // ប្រកាសអចេរណ៍ថា sum ជាបំនុលតត់
Char text ; // ប្រកាសអចេរណ៍ថា text ជាប្រភេទអក្សរ
boolean switch ;
byte x,y,z;
word data_bus;
long speed_of_light;
int mylist[10];

```

៦.១. ការប្រព័ន្ធឌែលបាន (Global Declaration)

អចេរដែលប្រកាសជាប្រភេទសកលអាចប្រើប្រាស់បានគ្រប់ទីកន្លែង (គ្រប់អនុគមន៍ដែលមានក្នុងកម្មវិធី) ។ ការប្រកាសអចេរជាសកលគឺត្រូវបានប្រកាសនៅពីរឬអនុគមន៍ void setup() ។

ឧចាងន៍ ៖

```

int pin; // ប្រកាសអចេរណ៍ថា pin ជាប្រភេទសកល
void setup()
{
    // program body;
}

```

៦.២. ការប្រព័ន្ធឌែលបានក្នុងតំបន់ (Local Declaration)

អចេរដែលប្រកាសក្នុងតំបន់អាចប្រើប្រាស់បានតែក្នុងអនុគមន៍ប៉ុណ្ណោះ ។ ការប្រកាសអចេរក្នុងតំបន់គឺត្រូវប្រកាសនៅក្នុងអនុគមន៍ void setup() ។

ឧចាងន៍ ៖

```

void setup()
{
    int pin ; // ប្រកាសអចេរណ៍ថា pin
    // អចេរដែលប្រើបានក្នុងតំបន់
    **** program body;
*****
}

```

៦.៣. ភារម្បទានននទេរក្រឹងStatement និងក្នុង

ភាសា C អនុញ្ញាតឱ្យប្រកាសប្រើនអចេរនៅក្នុងមួយ statement ឬន ។

ឧទាហរណ៍ ៖ int x , y , sum ;

តាមខាងក្រោមនេះបើយើងបានប្រកាសអចេរចំនួន ឬ គឺ អចេរ X ជាប្រភេទចំនួនគត់ អចេរ y ជាប្រភេទចំនួនគត់ និងអចេរ sum ជាប្រភេទចំនួនគត់ដីជំនួយ។

៦.៤. ភារម្បទានននទេរក្រឹង (Variable initialization)

Data_type Variable_name = value ;

ប្រភេទចំនួនយ លើយូវអចេរ = ៥៩២ ;

ឧទាហរណ៍ ៖ int LED =13;

Boolean LED_ON = false;

char mychar='A';

char mychar=35; // equal to 'A'

int LED1 [3]={8,9,10};

៧. PIN direction

ធាយសារកែវិន Digital របស់ Arduino មានទីសដោតីរ ពេលគីសដោច្បាលនិងចេញ ដូចនេះមុននឹងបើវិនិងណាមួយអោយវាដំណើរការអ្នកត្រូវប្រកាសទីសដោរបស់វាទាមុនសិន ហើយមាន Syntax របស់វាតី

Syntax:

pinMode (pin, Mode);

pin: មានន័យថាតីជាលេខដើរនិងDigitalរបស់Arduino ដើលអ្នកចង់អោយវាដំណើរការមួយណា

Mode : មានមានន័យថាអ្នកចង់អោយវាតាប្រភេទ(OUTPUT) ឬប្រភេទ(INPUT)

កែវិន Mode ចេញចេញជាប្រភេទ គឺ **OUTPUT , INPUT , INPUT_PULLUP** ។

OUTPUT : មានន័យថាប្រកាសឲ្យដើរនិងណាមួយជាដើរនិងប្រភេទ Signal Digital ។

INPUT : មានន័យថាប្រកាសឲ្យជាដើរនិងណាមួយជាដើរនិងប្រភេទ Signal Digital ។

INPUT_PULLUP : មាននៅយច្ចាប្រកាសឡើងណាមួយជានឹងប្រកបូល Signal Digital តែមានសុវត្ថិភាពដែលមិនមែនលទ្ធផលរបស់ខ្លួន នៅពីខាងក្រោមហើយជាប្រសិទ្ធភាព។

Example :

```
pinMode ( 13 , OUTPUT); // មាននំយច្ចាប់ឱកដើរទី13 ឲ្យដំណើរការជាថ្មី  
pinMode (12 , OUTPUT) ; // មាននំយច្ចាប់ឱកដើរទី12 ឲ្យដំណើរការជាថ្មី  
pinMode (3 , INPUT) ; // មាននំយច្ចាប់ឱកដើរទី3 ឲ្យដំណើរការជាបញ្ហាល
```

បសិនបើយើងកំណត់ដើម្បីជាបេកបេក នៅវានឹងមានការពិធីថ្ងៃទាំងមួយ ហើយភ្លាមជាក់

ធនធានក្នុងកម្រិតនេះអាចបញ្ចប់បើក LED (បើនឹងត្រូវបន្ថែមត្រួតពិនិត្យ) និងបិទបើកវិឡេចាន។

៤. អត្ថបន់យោងក្នុង

អត្ថន៍យោងក្នុងដែលពេងពេជ្យបន្ទាត់ពេលសរសរណ៍លើកម្មវិធី Arduino IDE សិក្សានៅលីបំពេះដើរ Arduino UNO (R3)

➤ **digitalWrite** (ការបញ្ជាក់សីញ្ញាល់ទិន្នន័យ)

របៀបដែល Arduino បញ្ជាសីឆ្លាប់ទៅក្រោដើម្បីបញ្ចារទៅកាន់អីមួយត្រា នានហេ
តាមរយៈ Syntax :

Syntax: **digitalWrite** (pin , value) ;

pin តីបញ្ញាក់ពីដើរលើកម្ពស់ UNO ដែលមានដើរលើ 0 ដល់ 13មានសំយចាថចង់ប្រើដើរមួយណាក់ចានក្នុងចំណោម 14 ដើរ ។

value មាននៅចាប់ពីលក្ខណៈ Digital ដែលត្រូវសរស់រ 1 (ភាគធំណើរការ) 0 (ភាគមិនធំណើរការ) បុក្ខសរស់រ HIGH (ភាគធំណើរការ) និង LOW (ភាគមិនធំណើរការ) ។

Example : `digitalWrite (12 , HIGH) ;` // កំណត់ដឹងទី 12 របស់ UNO មានជំនួយការ
`digitalWrite (10 , LOW) ;` // កំណត់ដឹងទី 10 របស់ UNO មិនជំនួយការ

➤ Delay Time

ការពន្លាបេល បូជ្ញាកដំណើរការ នៃព្រឹត្តិការណាមួយដែលកំពុងគីតឡើងនៅក្នុងកូដ របស់អ្នក ។ នៅក្នុង Arduino IDE , Delay ថែកចេញជាពីរ Syntax ៖

Syntax:

`delay (ms);`

`delayMicroseconds(us);`

ដែល ms: រយៈពេលគិតជាមីលីវិនាទី (1000 មីលីវិនាទី = 1 វិនាទី)

us: រយៈពេលគិតជាមីក្រុវិនាទី (1000 មីក្រុវិនាទី = 1 មីលីវិនាទី)

Example : `delay (1000); // ពន្លាបេល បូជ្ញាកដំណើរការ 1 វិនាទី`

`delayMicroseconds (1000); // ពន្លាបេលបូជ្ញាកដំណើរការ 1 មីលីវិនាទី`

➤ for loop statement

វាជាកនេរាមការងារមួយដែលធ្វើឡើងដែលរាយក្តឹងចំណាំនៃកំណត់លាមួយដែលចំណាំនៅក្នុងការងារ ជានេះអាចត្រូវបានគេកំណត់ថ្មីប៉ុណ្ណោះ បូរាប់ឡើង ។

Syntax: `for (Initialization; Expression; Increment / Decrement){`

`Statements;`

`}`

- **Initialization:** គឺជាកន្លែងដូចលើវត្ថុផ្លូវចាប់ផ្តើម
- **Expression:** គឺជាកន្លែងសម្រាប់កំណត់លំក្បែខ័ណ្ឌ
 - បើសិន Expression ពិភាក្សានិងអនុវត្តន៍កាលបណ្តាល Statements ដែលស្ថិតនៅក្រោមឱវាទរបស់វា
 - បើសិន Expression មិនពិភាក្សានិងមិនអនុវត្តន៍កាលបណ្តាល Statements ដែលស្ថិតនៅក្រោមឱវាទរបស់វា
- **Increment / Decrement:** គឺជាកន្លែងសម្រាប់កំណត់តម្លៃកើន បុ ចូយ។

ឧបាទេន៍ ១ ៖

```
for ( int I = 0 ; I <= 99 ; I++) {  
    println ( I ) ; // prints 0 to 99  
}
```

ឧបាទេន៍ ២ ៖

```
void setup () {  
    for ( int x=2 ; x <=9 ; x++ )  
    { pinMode ( x , OUTPUT); }  
}  
  
void loop () {  
    digitalWrite ( 2 ,HIGH);  
    digitalWrite ( 3 ,HIGH);  
    digitalWrite ( 4 ,LOW);  
    digitalWrite ( 5 ,HIGH);  
    digitalWrite ( 6 ,LOW);  
    digitalWrite ( 7 ,HIGH);  
    digitalWrite ( 8 ,LOW);  
    digitalWrite ( 9 ,HIGH);  
}
```

➤ Array type

Array type គឺជាការប្រមូលផ្តើនៃអចេរប្រើប្រាស់ក្នុងកន្លោមទំនួយហើយត្រូវបានគេបែងកប្រើរិញ្ជ តាមរយៈការបែងចាត់ វិសន្ទុទស របស់វា ។ ឧងតាមជាមុនបានណែនាំមួយចំនួន ៖

Example 1 : int a [5] ;
 Int a [] = { 3 , 4 , 6 , 8 , 10 , 12 };

Example 2: int x [5] = { 3 , 5 , 7 , 9 , 10 } ;

Example 3: int dig1 [7] = { 2 , 3 , 4 , 6 , 8 , 10 , 12 };
 Int a =dig1 [4] ; // មានន័យថា a = 6

➤ **Macro**

Macro គឺជារបៀបបាន បុ ក៏ណាត់អីមួយទុកជាមុន ។

Syntax:

#define constantname value

ដែល **constantname** គឺជាលេខា៖ណាមួយដែលចង់កំណត់ ,

Value គឺជាដំឡើរបស់ **constantname**

Example :

```
#define      write   printf
#define      read    scanf
#define      Begin   void main(){}
#define      End    }
#define mydelay delay(100)
#define dooropen digitalWrite(5,HIGH )
```

ເຮັດວຽກຂອງລົງທະບຽນ ພະຍາຍາ ປະເທດລາວ

១. សំខ្លែង (Constants)

Constants គឺជាលេខ ប្រអក្សរ (number or character) ដែលមានតម្លៃមួយចំនួនភាព ផ្តាស់ប្តូរតម្លៃបាននៅក្នុងកម្មវិធី។ constants គឺត្រូវបានផ្តើកនៅក្នុង flash program memory នៃ microcontroller ។នៅក្នុងកម្មវិធី C constants មានជោចជាទិ integer constants, Floating Point constants, character constants, string constants, Enumerated constants ។

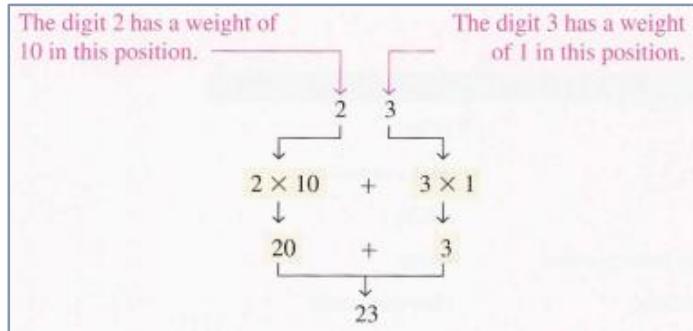
9.9. integer constants

Integer constants គឺជាចំនួនតម្លៃចំនួនដូចជាលេខគោលដៅ (decimal number) លេខគោលដៅប្រាំមួយ (hexadecimal number) លេខគោលប្រាំបី (octal Number) លេខគោលពីរ (binary number) ។ Compiler អាចស្វាត់ពីប្រព័ន្ធគោលបានអាស៊យទេឡើ បុរិបទដែលមាន ហិរញ្ញវត្ថុមួយណាមួយ ។

Format	Prefix	Example
Decimal		Const Max = 100
Hexadecimal	0x or 0X	Const Max = 0xFF
Octal	0	Const Max = 016
Binary	0b or 0B	Const Max = 0b11110000

❖ ଦେଶାଂକ୍ରମିତି (decimal number)

Decimal Number គឺជាលេខប្រព័ន្ធគោលដៅ ដែលមានលេខ 0.1.2.3.4.5.6.7.8និងលេខ១០ នៅក្នុងប្រព័ន្ធ Decimal គឺមាន១០តួលេខបី នៃតួរាយចិន មែនមាននំយ៉ាហើយឯងកប់បាន តែ១០តួលេខនោះទេ គឺយើងកប់បានច្រើនដោយគ្រាន់តែធ្វើការបន្ទូមនៅទីតាំងនៅចំណុះនៅ។



❖ លេខលោនតី (binary number)

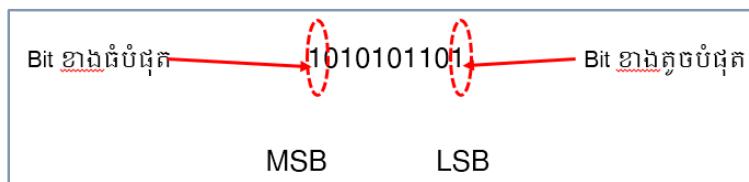
- គារប័ណ្ណBinary (Counting in Binary)
ដើម្បីធ្វើការរាប់ចំនួនBinary គឺយើងធ្វើការបែន្ទែមនូវទីតាំងផុកចុកងប្រព័ន្ធគោល10 ដែរ (Decimal systems) ដោយគ្រាន់តែយកពីរធ្វើជាគោលដោយចាប់ផ្តើម 0 និង 1 ។

Decimal Number	Binary Number
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

តាមរយៈតារាងខាងលើយើងយើងប្រសិនបើយើងមានចំនួន Binary ចំនួន 4 bits នៅ៖យើងអាចរាបចំនួន Decimal ពី 0 ដល់ 15 ។ ជាទុទៅប្រសិនបើយើង មានចំនួន n bit នៅ៖យើងអាចរាបចំនួនពី 0 ដល់ $2^n - 1$ ។

ឧទាហរណ៍ បើ $n=5$ នៅ៖យើងត្រូវ រាបចំនួនពី 0 ដល់ $(2^5 - 1) = 31$ ។

ចំនួន Binary ត្រូវគោននកំណត់ទីតាំងរបស់វា(Weight) មានពីរប្រភេទ គឺ Bit ដែល នៅខាងស្តាំគឺចំនួនបំផុតហេរិថា LSB (least significant bit) និង weigh របស់វា ត្រូវធ្វើនៅ ជាលំដាប់ពីស្តាំទៅឆ្នៀងតាមរយៈស្ថោយគុណ នៃ 2 ($2^0, 2^1, 2^2, 2^3, \dots, 2^n$) ហើយ bit ដែលនៅផ្លូវខាងឆ្នៀងបំផុតគឺហេរិថា MSB (most significant bit) ។ ចំពោះ ចំនួន កៅំសត្រូវបានគេប្រើដឹងដែរនៅក្នុងប្រព័ន្ធបinary ដែល weight ចិញ្ចុះពីឆ្នៀងទៅ ស្តាំតាមរយៈស្ថោយគុណនៃ 2 ដែល ។



❖ លេខសោនជចំប្បែកចំនួន (hexadecimal number)

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A

11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

❖ លេខតម្លៃទូទៅ (Octal Number)

ប្រព័ន្ធប័ត្រន Octal គឺជាប្រព័ន្ធប័ត្រនមួយដែលមានប្រាំបីតូលេខ តី 0,1,2,3,4,5,6,7 ។ ដើម្បីធ្វើការរាប់ចំណុនឈ្មោះបានលើសពីប្រាំពីរ គឺយើងត្រូវធ្វើការបន្ថែមនូវទីតាំងនៃចំណុនមកដូចការងារ ដើម្បីកំណត់ចំណុនលើសពីរ ។ ដើម្បីកំណត់ចំណុនលើសពីរ គឺគឺជាប្រព័ន្ធដឹបតុលេខ Decimal គឺគឺជាប្រព័ន្ធដឹបតុលេខ 8 [(15)₈] ។ ដោយសារតែចំណុន Octal មួយ 1 digit ស្ថិតិនឹង 3 bits នៃចំណុន Binary ដូចនេះវាមានភាពងាយស្រួលក្នុងការបំលែងពី Octal to Binary ដោយតូលេខ Octal និមួយស្ថិតិនឹងតូលេខ Binary ដូចតារាងខាងក្រោម ។ ដោយសារតែចំណុន Octal មួយ 1 digit ស្ថិតិនឹង 3 bits នៃចំណុន Binary ដូចនេះវាមានភាពងាយស្រួលក្នុងការបំលែងពី Octal to Binary ដោយតូលេខ Octal និមួយស្ថិតិនឹងតូលេខ Binary ដូចតារាងខាងក្រោម ។

Octal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

១.២. Floating Point Constants

Floating Point Constants គឺជាប័ត្រនពិតចេរ ។

ឧត្តមាឌ់៖

```
cont T_MAX = 32.60 ;           // declare temperature T_MAX
cont T_MAX = 3.260E1 ;         // declare the same constant T_MAX
```

១.៣. Character Constants

Character Constants គឺជាអក្សរដែលនៅចន្ទោះពី single quotation marks ('.....') ។

ឧត្តមាឌ់៖

```
cont I_class = 'A' ;           // declare constants I_class
cont II_class = 'B' ;          // declare constants II_class
```

១.៤. String Constants

String Constants គឺជាអក្សរដែលនៅចំណោមពី double quotation marks (".....") ។

ឧទាហរណ៍ :

```
const Message_1 = " Press the START button" ; // message 1 for LCD
const Message_2 = " Press the RIGHT button" ; // message 2 for LCD
const Message_3 = " Press the LEFT button" ; // message 3 for LCD
```

១.៥. Enumerated Constants

Enumerated Constants គឺជាប្រភេទចំណួនគត់ហើយត្រូវបានគេប្រើប្រាស់ដើម្បីបង្កើតកម្មវិធីឱ្យមានការងាយស្រួលការទាញយកតម្លៃតាមលំដាប់ដែលតម្លៃរបស់ពីលេខស្តីឱ្យទៅ ។

ឧទាហរណ៍ :

```
enum colors {black , brown, red, orange, yellow};
```

តាមខាងក្រោមនេះយើងអាចទទួលបានតម្លៃចំណាត់ការក្នុងក្រោម៖

black ស្តី 0 , brown ស្តី 1, red ស្តី 2, orange ស្តី 3, yellow ស្តី 4

Led= black; // បញ្ចូលតម្លៃ 0 ឱ្យទៅអារ៉ូ led

Led = 0; // បញ្ចូលតម្លៃ 0 ឱ្យទៅអារ៉ូ led

២. ក្រុមការណិតិ (Operators)

២.១. ក្រុមការណិតិធម្មតា (Arithmetic Operators)

សញ្ញា	អត្ថន័យ	ឧទាហរណ៍
+	បុក	a+b
-	ដក	a-b
*	គុណ	a*b
/	ថែក	a/b
%	ថែកយកសំណាល់	a%b

/* ការបុកពីចំណួនគត់ */

5 + 12 // ស្តី 17

3.1 + 2.4 // ស្តី 5.5

```

/* ការដែកពីចំនួនគត់ */
120 - 5      // ស្មើ 115
10 - 15     // ស្មើ -5
/* ការគុណពីរចំនួនគត់ */
12 * 3      // ស្មើ 36
2 * 5      // ស្មើ 10
/* ការចែកពីរចំនួនគត់ */
12 / 3      // ស្មើ 4
5 / 3      // ស្មើ 1
/* ការចែកយកសំណាល់ */
7 % 3      // ស្មើ 1
5 % 3      // ស្មើ 2

```

២.២. រៀងរាល់សំណើសង្គម (Relational Operators)

សញ្ញា	អត្ថន័យ	ឧទាហរណ៍
>	ជំជាង	$a > b$
\geq	ជំជាង បុស្មើ	$a \geq b$
<	តូចជាង	$a < b$
\leq	តូចជាង បុស្មើ	$a \leq b$
\equiv	ស្មើគ្នា	$a \equiv b$
\neq	ខ្ពស់	$a \neq b$

ឧទាហរណ៍ បើ $x = 10$

```

x > 8      // លទ្ធផលគឺពិត(1)
x == 10    // លទ្ធផលគឺពិត(1)
x < 100    // លទ្ធផលគឺពិត(1)
x > 20     // លទ្ធផលគឺមិនពិត(0)
x != 10    // លទ្ធផលគឺមិនពិត(0)
x >= 10   // លទ្ធផលគឺពិត(1)
x <= 10   // លទ្ធផលគឺពិត(1)

```

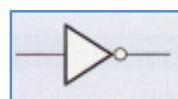
၂.၃. ဗုဒ္ဓဘာသီခြား Boolean (Boolean Operator)

ត្រូវបានប្រើប្រាស់សម្រាប់តែស្ថិតិមិនុយិលលក្ខខណ្ឌពីវិធីគ្រឿនតាមការគណនាកត្តិទោរ។ វាមានលក្ខណៈខាងក្រោម៖

សញ្ញា	អត្ថន័យ	ឧទាហរណ៍
!	ប្រមាណវិធី Not gate	$\neg a$
&&	ប្រមាណវិធី AND gate	$a \& b$
	ប្រមាណវិធី OR gate	$a \mid\mid b$

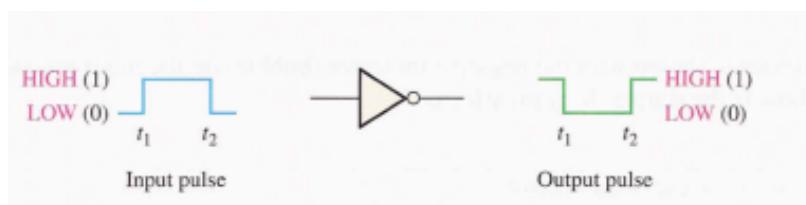
- ❖ ក្រុងតាមទិន្នន័យ Not gate (The Inverter)

Logic symbol



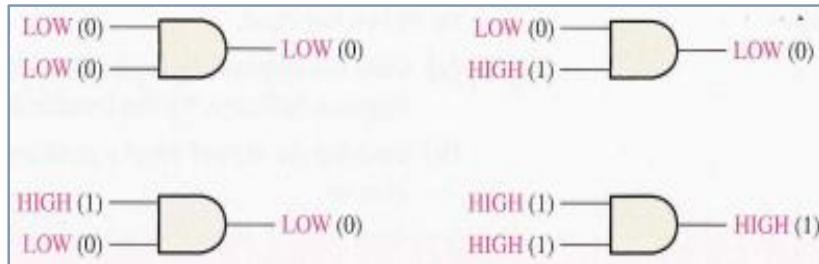
និមិត្តសញ្ញា Not gate គឺជាការបង្ហាញជាសញ្ញា រដ្ឋម៉ោង (0) ដែលបង្ហាញពីរការ បំលែងវិភាគដាក់ Complement ដែលគេយើងមាននៅផ្លូវក្រច្ចកថ្មី និងផ្លូវក្រច្ចកថ្មីល នៃនិមិត្ត របស់វា ។ ជាទុទេវប្រកបច្ចូលគឺនៅផ្លូវក្រខាងឆ្វេងហើយប្រកបថ្មីនៅផ្លូវក្រខាងស្តាំ នៅពេលគេដាក់សញ្ញារដ្ឋម៉ោងនៅប្រកបច្ចូល នៅពេលហេចចាត់ Active low Input ប្រសិន ហើសញ្ញាបដ្ឋម៉ោងត្រូវបានគេដាក់ នៅខាងប្រកបថ្មី គេហេចចាត់ Active low output ។

Input (a)	Output (!a)
LOW (0)	HIGH (1)
HIGH (1)	LOW (0)



❖ ប្រព័ន្ធលទ្ធផលី AND gate

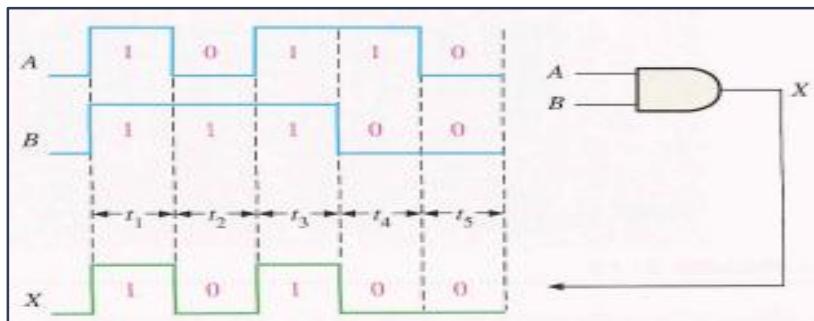
លក្ខណៈ: ដំនើរការ និមិត្តសញ្ញា និងតារាងធាតុពិត (1 = HIGH ; 0 = LOW)



តារាងធាតុពិត សម្រាប់ប្រព័ន្ធបូលពីរ

ប្រព័ន្ធបូល (Input)		ប្រព័ន្ធបូល (Output)
A	b	a&&b
0	0	0
0	1	0
1	0	0
1	1	1

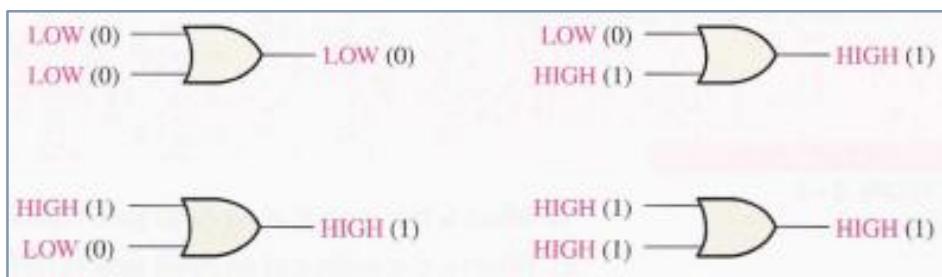
ក្របដំណើរការ សម្រាប់ប្រព័ន្ធបូលពីរ



❖ ប្រព័ន្ធលទ្ធផលី OR gate

លក្ខណៈ: ដំនើរការ និមិត្តសញ្ញា និងតារាងធាតុពិត (1 = HIGH ; 0 = LOW)

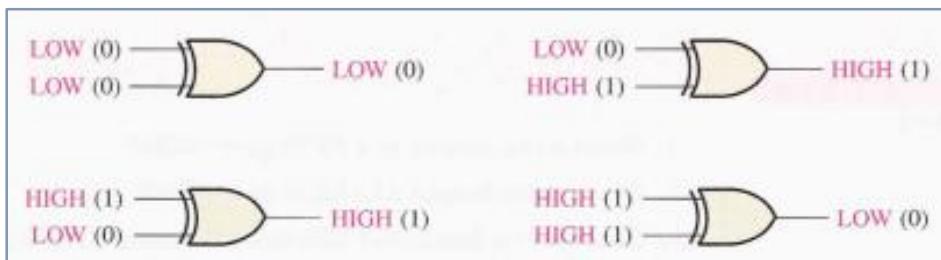
តារាងធាតុពិត សម្រាប់ប្រព័ន្ធបូលពីរ



ប្រភពិល (Input)		ប្រភពិល (Output)
A	b	$a \parallel b$
0	0	0
0	1	1
1	0	1
1	1	1

❖ ប្រភពិល XOR gate

លក្ខណៈ: ដំនើរការ និមិត្តសញ្ញា និងតារាងធាតុពិត (1 = HIGH ; 0 = LOW)



តារាងធាតុពិត សម្រាប់ប្រភពិល

ប្រភពិល (Input)		ប្រភពិល (Output)
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

ឧទាហរណ៍៖

a	b	!a	!b	a&&b	!(a&&b)	a b	!(a b)
0	0	1	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	0	1	0	1	1	0
1	1	0	0	1	0	1	0

២.៥. ប្រទល់លក្ខណីនៃ Bit (Bitwise Operators)

ជាអនុគមន៍តក្ខវិទ្យាដែលរួមមាន៖ AND , OR , NOT , XOR ,shift left ,shift right ,.....។
ប្រតិបត្តិការទាំងនេះធ្វើឡើងតែនៅលើតំបន់ bit ហើយវាបានប្រភេទ Logic gate ។

សញ្ញា	អត្ថន័យ	ឧទាហរណ៍
&	ប្រមាណវិធី AND ចំពោះ bit នឹមួយា	$a \& b$
	ប្រមាណវិធី OR ចំពោះ bit នឹមួយា	$a b$
^	ប្រមាណវិធី XOR ចំពោះ bit នឹមួយា	$a ^ b$
<<	វគ្គលដ្ឋាន (shift left)	$a << b$
>>	វគ្គលស្តាំ (shift right)	$a >> b$
~	ប្រមាណវិធី NOT ចំពោះ bit នឹមួយា	$\sim a$

ឧទាហរណ៍៖ បើ $a = 7$; $b = 4$; (a និង b មានទំហំ 8-bit)

គណនា $a \& b$, $a | b$, $a ^ b$, $a << 2$, $b >> 1$, $\sim a$, $\sim b$

យើងត្រូវបង្កើតលេខគោល10 ទៅលេខគោល2 សិនដើម្បីងាយស្រួលដោះស្រាយ

$$a = 00000111_2 \quad b = 00000100_2$$

ប្រមាណវិធី	លទ្ធផល	
$a \& b$	$a = 00000111$	00000100
$a b$	$a = 00000111$	00000111
$a ^ b$	$a = 00000111$	00000011
$a << 2$	$a = 00000111$	00011100
$b >> 1$	$b = 00000100$	00000010
$\sim a$	$a = 00000111$	11111000
$\sim b$	$b = 00000100$	11111011

២.៥. ប្រអប់និងផ្តល់សង្គម (Assignment Operators)

Assignment Operators គឺជាប្រមាណវិធីសម្រាប់ផ្តល់តម្លៃទ្វាមចំណាំ និងជាប្រមាណវិធីតណិតវិញ្ញាដែលត្រូវបានគេបង្កើតដើម្បីសរស់រួចរាល់លើវិធីសម្រាប់គោលដៅ ហើយ វាបានប្រមាណវិធីបណ្តាក់ត្រាបើអចេរតម្លៃយ។

សញ្ញា	អត្ថន័យ	ឧទាហរណ៍
=	ប្រមាណវិធីផ្តល់តម្លៃ	$a = 1$
+=	ប្រមាណវិធីបន្ថែមតម្លៃពីលើតម្លៃចាស់	$a += 1$
-=	ប្រមាណវិធីបន្ថែមឬបន្ទាយតម្លៃពីលើតម្លៃចាស់	$a -= 2$
/=	ប្រមាណវិធីយកតម្លៃចាស់មកចំក	$a /= 3$
*=	ប្រមាណវិធីយកតម្លៃចាស់មកគុណ	$a *= 2$
++	ប្រមាណវិធីបន្ថែមតម្លៃមួយពីលើតម្លៃចាស់	$a ++ \text{ or } ++ a$
--	ប្រមាណវិធីបន្ថែមឬបន្ទាយតម្លៃមួយពីលើតម្លៃចាស់	$a -- \text{ or } -- a$

ឧទាហរណ៍៖ តើមាន $a = 8$;

ប្រមាណវិធី	លទ្ធផល
$a = 8$	$a = 8$
$a += 1$	$a = 9$
$a -= 2$	$a = 6$
$a /= 2$	$a = 4$
$a *= 2$	$a = 16$
$a ++$	$a = 9$
$a --$	$a = 7$

ចំពោះ $a += 1$ គោរពស្រស់រក្សាសំរាប់ $a = a + 1$

$$a = 8 + 1 = 9$$

២.៦. លទ្ធផលនៃបញ្ហាសញ្ញាប្រព័ន្ធទីផ្លូវ

ល.រ	បណ្តាលសញ្ញាប្រព័ន្ធទីផ្លូវ	លំដាប់ប្រមូលដ្ឋាន
1	([])->	ពីធ្វើឱ្យទៅស្តាំ
2	!~++--+(unary)-(unary)*Pointer&Pointer	ពីស្តាំទៅធ្វើឱ្យ
3	*/%	ពីធ្វើឱ្យទៅស្តាំ
4	+-	ពីធ្វើឱ្យទៅស្តាំ
5	< >	ពីធ្វើឱ្យទៅស្តាំ
6	<< = > > =	ពីធ្វើឱ្យទៅស្តាំ
7	= = !=	ពីធ្វើឱ្យទៅស្តាំ
8	&	ពីធ្វើឱ្យទៅស្តាំ
9	^	ពីធ្វើឱ្យទៅស្តាំ
10		ពីធ្វើឱ្យទៅស្តាំ
11	&&	ពីធ្វើឱ្យទៅស្តាំ
12		ពីធ្វើឱ្យទៅស្តាំ
13	?:	ពីស្តាំទៅធ្វើឱ្យ
14	= += -= *= /= &= ^= = <= >=	ពីស្តាំទៅធ្វើឱ្យ

ការណែនាំបណ្តាលសញ្ញាប្រព័ន្ធទីផ្លូវខាងលើដែលសរសេរនៅលើបន្ទាត់ព័ម្ធយនោះបាននៅក្នុងមានអាជិភាពស្តីពី និងបណ្តាលសញ្ញាដែលបិតនៅបន្ទាត់ខាងលើ មានអាជិភាពជាងបណ្តាលសញ្ញាដែលបិតនៅបន្ទាត់ខាងក្រោម ។

ឧទាហរណ៍ 1: បើ $a = 1$

```
b = 3 * ++a      // ដោយសញ្ញា ++ មានអាជិភាពជាងសញ្ញា * ដំបូង
                  // យើងគណនា ++a សិនបន្ទាប់មកទីបន្ទាត់មានក្រោម
= 3 * (2)
= 6
```

ឧទាហរណ៍ 2: គោលកនេរក្រម $8/4*6$

$$2 * 6 = 12$$

ដោយសញ្ញា / មានអាជិភាពស្តីសញ្ញា * បីទីនៅលើដាប់នៃការគណនាទីធ្វើឱ្យទៅស្តាំ ។

ដូចនេះយើងគ្ររគណនា $8/4$ សិនហើយទីបន្ទាត់មែន 2 ទីបន្ទាត់និង 6 ។

ទេសកូមិកភាពពេជ្តាក់សនស្តាយរដ្ឋមន្ត្រី (Control Statements)

នៅក្នុងភាសាកម្មវិធី C Control Statement មានសារ៖ សំខាន់សម្រាប់ធ្វើការត្រួតពិនិត្យទៅលើលំហ៊ុវន់ដំណើរការរបស់កម្មវិធីទៅតាមលំដាប់លំដាយត្រឹមត្រូវយកើយលើក្នុងខ្លួនដែលទទួលនូវតម្លៃពីរ គឺ មួយពិត(True) ឬ មួយមិនពិត(False) ។

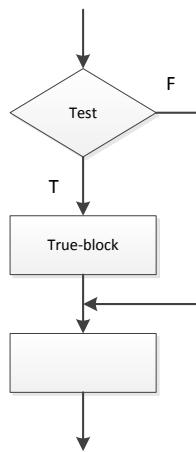
Control Statements មានដូចជា៖

- ❖ if control statements
- ❖ switch case control statements
- ❖ while loops
- ❖ do-while
- ❖ go to statements
- ❖ for statements
- ❖ break statements
- ❖ continue statements
- ❖ Function (return)

១. ឥឡូវ if control statements

➤ **if (conditional) and ==, !=, <, > (comparison operators)**

if ត្រូវបានប្រើសម្រាប់ប្រពិបត្តិការប្រើប្រាស់លក្ខណៈ ឬ ប្រចើនជាមួយគ្នា ។
បើលក្ខណៈពិតនោះវារាយដុលអ្នកសរស់រចនា ។



Syntax :

```
if (someVariable > 50)
{
    // do something here
}
```

ឧទាហាននេះ :

- ❖ របៀបទី១៖

```
if (x > 120)
    digitalWrite(LEDpin, HIGH);
```

- ❖ របៀបទី២៖

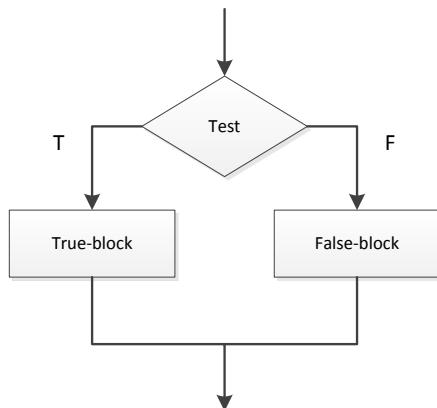
```
if (x > 120){ digitalWrite(LEDpin, HIGH); }
```

- ❖ របៀបទី៣៖

```
if (x > 120){
    digitalWrite(LEDpin1, HIGH);
    digitalWrite(LEDpin2, HIGH);
}
// all are correct
```

➤ if / else

គ្រឿងបញ្ជីតាមឱ្យ if ដែលលទ្ធផលរបស់វាធ្វើការនៅពេលលក្ខខណ្ឌ នៅក្នុង if មិនពិត
នៅវានឹងលោកទៅធ្វើក្នុងក្នុងរបស់ else ។



Syntax :

```
if (someVariable < 500)
{
    // action A
}
else
{
    // action B
}
```

ឧទ្ធសាស្ត្រ

```
if (pinFiveInput < 500)
{
    // do Thing A
}

else if (pinFiveInput >= 1000)
{
    // do Thing B
}

else
{
    // do Thing C
}
```

೭. ಇಂತಹ switch case control statements

switch case control statements ស្របដៃងត្នានីង if ដើរ ពេកវាមានជំណើករាជទឹសត្នាត្រង់កយកតម្លៃនៃអចេរម្បយទៅធ្វើបនិងតម្លៃជាប្រចិននោះហើយវាគ្មែលក្នុងណាកុងត្រង់ចំនួចនោះថមុន ។ ជាម្បយត្នានេះដើលពាក្យតនឹង៖ break ត្រូវបានប្រើប្រាស់ដើម្បីបញ្ចប់លក្ខខណ្ឌនៅក្នុងករណីម្បយ។ របស់តម្លៃកំណត់នោះក្រោយពេលពេលវាបានធ្វើការដោយសារតម្លៃដៃងត្និនីកតំលៃរបស់វាធ្វើបនិងតម្លៃលែអចេរម្បយ គឺត្រូវត្រូវ ។

Syntax :

```

switch (var) {
    case label:
        // statements
        break;
    case label:
        // statements
        break;
    default:
        // statements
        break;
}

```

- **var** : ជាមធ្យរដើម្បីរបស់វាត្រូវបានផ្តល់បន្ទីនៅពេលវិភាគឱ្យមួយទៅ
- **label** : គម្រោងដើម្បីរបស់វាដែលត្រូវបានផ្តល់បន្ទីនៅពេលវិភាគឱ្យមួយទៅ

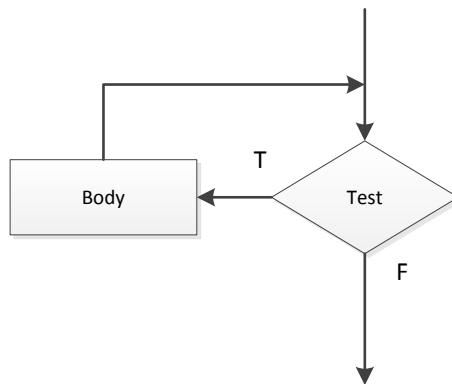
ឧបាទេរណ៍៖

```

switch (var) {
    case 1:
        //do something when var equals 1
        break;
    case 2:
        //do something when var equals 2
        break;
    default:
        // if nothing else matches, do the default
        // default is optional
        break;
}

```

၃. ဇုန်ခံ While () Loops structure



Syntax :

```
while(expression){  
    // statement(s)  
}
```

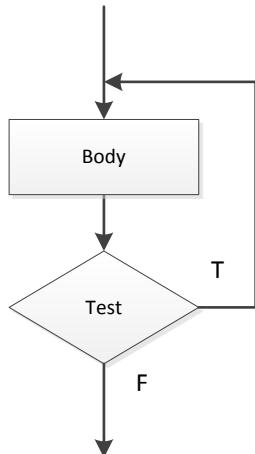
- បើសិន expression ពិតវានិងអនុវត្តន៍កាល់បណ្តាល Statements
ដែលស្ថិតនៅក្រោមឱ្យរបស់វា
 - បើសិន expression មិនពិតវានិងមិនត្រូវអនុវត្តន៍កាល់បណ្តាល Statements
ដែលស្ថិតនៅក្រោមឱ្យរបស់វានេះ

ឧទាហរណ៍

```
var = 0;  
  
while(var < 200){  
    // do something repetitive 200 times  
  
    var++;  
  
}
```

៤. ច្បាស់ do...While () Loops structure

ត្រូវបានគេយកមករបីនៅពេលណាដែល មានការអនុវត្តន៍ការងារបានមួយដែងយ៉ាងហេចច្បាស់បានក្នុងក្នុងផ្ទាល់ក្នុងខ្លួនគ្នាបាយ នៅវីបច្ចុប្បន្នទៅការកំណត់ do ... while() Loop មករបី។



Syntax:

```

do{
    Statements;
} while(test condition );
  
```

- បើសិន test condition ពិតវានិងអនុវត្តន៍ការលែបណ្តាល Statements ដែលស្ថិតនៅក្រោមឱ្យទរបស់វាបន្ទាប់។
- បើសិន test condition មិនពិតវានិងបញ្ចប់ការអនុវត្តន៍ការលែបណ្តាល Statements ដែលស្ថិតនៅក្រោមឱ្យទរបស់វា។

ឧទាហរណ៍៖

```

do
{
    delay(50);      // wait for sensors to stabilize
    x = readSensors(); // check the sensors
}
while (x < 100);
  
```

៥. ឧប្បជ្ជតែល់ goto Statement

goto statement គឺជាមួយបញ្ហារប្រើសម្រាប់អោយកម្មវិធីលោកស្របនៃលងទៅទីតាំងណាមួយដោយគ្មាន លើក្នុងណាគាមរយៈណ៍ៗបន្ថែមនៃ Label ។

Syntax: Label_Name :

...

 goto Label_name;

or

 goto Label_Name;

...

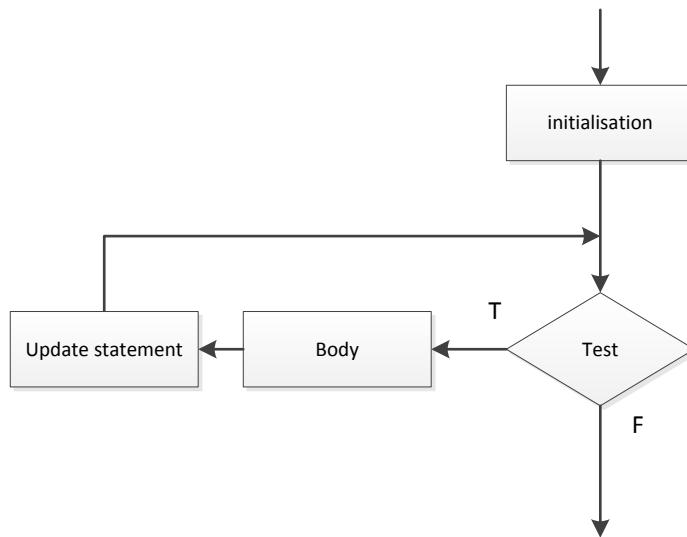
 Label_Name :

ឧបាទាន់នេះ :

```
for(byte r = 0; r < 255; r++){
    for(byte g = 255; g > -1; g--){
        for(byte b = 0; b < 255; b++){
            if (analogRead(0) > 250){
                goto bailout;
            }
            // more statements ...
        }
    }
}
bailout:
```

៦. ឧប្បជ្ជតែល់ for Statement

គឺត្រូវបានគេយកមកប្រើនៅពេលណាដែល គេដឹងនៅចំនួនដឹងនៃការអនុវត្តន៍ច្បាស់លាស់ ក្រាមលើក្នុងណាគាមកោដ្ឋាកែសនស្តាយ នៅ៖ទីបគេយកទៅនៃ for(...) Loop មកប្រើ។

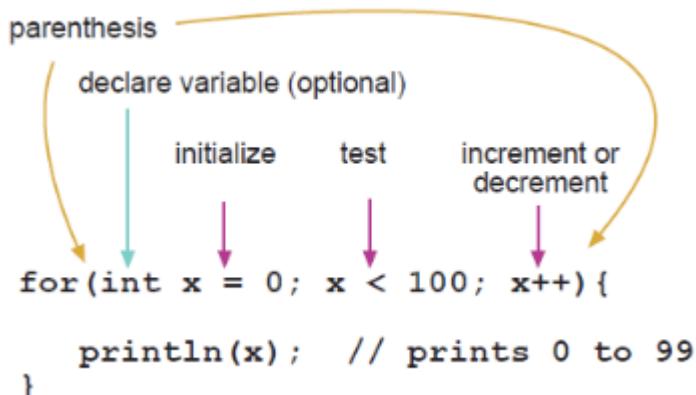


Syntax:

```

for (initialization; condition; increment) {
    //statement(s);
}
  
```

ក្បាល់ក្រោមបង្ហាញពីដំណើរការនៃលក្ខណៈ for loop ធ្វើការនេះ



ឧត្ថម្រន៍ 1 :

```

// Dim an LED using a PWM pin
int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10
void setup()
{
  
```

```

        // no setup needed
    }

    void loop()
    {
        for (int i=0; i <= 255; i++){
            analogWrite(PWMpin, i);
            delay(10);
        }
    }

```

ឧទាហរណ៍ 2 ៖

```

void loop()
{
    int x = 1;

    for (int i = 0; i > -1; i = i + x){

        analogWrite(PWMpin, i);

        if (i == 255) x = -1;          // switch direction at peak

        delay(10);
    }
}

```

៣. ច្បាប់ break Statement

break Statement គឺជា Keyword មួយដែលត្រូវប្រើសម្រាប់ចាកចេញពី Loop ។

ឧទាហរណ៍ ៣ ៖

```

for (x = 0; x < 255; x ++)

{
    analogWrite(PWMpin, x);

    sens = analogRead(sensorPin);

    if (sens > threshold){      // bail out on sensor detect

```

```

x = 0;

break;

}

delay(50);

}

```

៤. ច្បាប់ continue statements

continue Statement គឺជា Keyword មួយដែលគោរពប្រើសម្រាប់បន្ថាការអនុវត្តន៍ Loop ឡើងវិញ ។
ឧបាទោន់ ៖

```

for (x = 0; x < 255; x ++)

{   if (x > 40 && x < 120){    // create jump in values

        continue;

}

analogWrite(PWMpin, x);

delay(50);

}

```

៥. ច្បាប់ Function (return)

Functions គឺជាបណ្តុះក្នុងដែល User សរស់រឡើងដើម្បីទ្វាគនុវត្តន៍នូវការងារណាមួយ
ដែល User ចង់បាន។ នៅក្នុងភាសា C យើងអាចបង្កើត Functions បានប្រើនៅក្នុងការងារ
ដែលយើងចង់ដោះស្រាយ ដោយយើងត្រូវបង្កើតជាបច្ចុប្បន្ននឹងក្រុមហ៊ុន។ នៅពេល យើង បង្កើត
Function យើងត្រូវធ្វើការប្រាប់ទៅ Compiler ថា Function យើងលើប្លែងមីនី មាន Return type
ដែរបុទ្រ និងមាន Parameters ដែរបុទ្រ។

Syntax:

```

return_type function_name( parameter list ){

    body of the function

}

```

⊕ Return type គឺជាតម្លៃដែល Function និងត្រូវរាយឡើង បើសិនជា Function

មិនមាន Return តម្លៃ គោរព keyword void ។

- + Function name គឺជាលេខ្លោះរបស់ Function និមួយ។
 - + Parameters គឺជាអចេរ ដែលយើងនឹងបានក្នុងអាយុយទៅ Function។
 - + Function body គឺជាឫ្សែនរបស់ Function ដែល Function នៅក្នុងត្រូវបានអាយុយទៅ។

```
void function_name() {  
    ...  
    ...  
}  
  
int main() {  
    ...  
    ...  
    function_name();  
    ...  
}
```

ចំណាំ: ប្រសិនបើយើង បង្កើត Function នៅពីលី main program នោះយើងមិនចាំបាច់យើងការប្រកាសលួច: Function នៅនៃលី main program ទេ។ តែដូរយកទៅវិញ បើយើងបានបង្កើត Function នៅពីខាងក្រោម main program នោះយើងត្រូវរៀបចំការប្រកាស លួច:របស់ Function នៅពីលី main program ហើយត្រូវជាក់ Parameter គ្មាយដូចទៅនឹង Parameter នៅក្នុង Function ដ៏ខ្លួន។

ଓଡ଼ିଆ

```
int checkSensor(){  
    if (analogRead(0) > 400) {  
        return 1;  
    }  
    else{  
        return 0;  
    }  
}
```

```
}

void loop(){
    // brilliant code idea to test here
    return;
    // the rest of a dysfunctional sketch here
    // this code will never be executed
}
```

➤ សារៈសំខាន់នៃការប្រើប្រាស់ Function

- ងាយស្រួលដល់ការសរស់រ Function តួចុងបានត្រឹមត្រូវ
- កាត់បន្ថយការសរស់រ Code ដែលបាន
- អាចប្រើបានប្រើប្រាស់បាន
- ផ្តល់ភាពងាយស្រួលដល់ការអាជីវកម្ម សរស់រ និងកែត្រា

အော်လုပ်ဆိုင် နှင့် Serial Monitor နည်းလမ်း

9. Serial Monitor

- Serial protocol គឺជាកញ្ចប់ទិន្នន័យត្រូវបានប្រព័ន្ធគោលដៅ ក្នុងរយៈពេលម៉ោងមួយក្នុងរយៈពេលម៉ោង។ រយៈពេលម៉ោងនេះគឺជាលើករាបច្បាស់ទិន្នន័យគឺជាតុលាកិតជាទុលាកិត bits per second (baud rate)។ ស្ថិតិថ្លែង baud rate មានដូចជា 300 , 600 , 1200 , 2400 , 4800 , 9600 , 14400 , 19200 , 28800 , 38400 , 57600 និង 115200 ។
 - Serial of Arduino

➤ Serial of Arduino

Arduino IDE ដែលយើងប្រើមានផ្ទាប់មកជាមួយនូវ Serial Monitor ដែលអាចឲ្យយើងធ្វើពេតស្ថាប្រើប្រាស់ជាមួយនឹង serial coder របស់វា បុក្រីត ផ្តាប់ជាមួយខ្លួន ដើម្បីពន្លាបាន។
ជាពីរទី Serial Monitor របស់ Arduino ត្រឹមសរសរនិងជ្រើសរើស baud rate 9600។
ដើម្បីអាចប្រើប្រាស់វាតាមយើងត្រូវយល់ដឹងអំពី Function of syntax មួយចំនួនរបស់មានដូចជា៖

- ❖ if (Serial)
 - ❖ available()
 - ❖ availableForWrite()
 - ❖ begin()
 - ❖ end()
 - ❖ find()
 - ❖ findUntil()
 - ❖ print()
 - ❖ println()
 - ❖ read()
 - ❖ readBytes()
 - ❖ readBytesUntil()

- ❖ `readString()`
- ❖ `readStringUntil()`
- ❖ `setTimeout()`
- ❖ `write()`
- ❖ `serialEvent()`

➤ **`begin()`** :

បើកអោយដើង serial port របស់ អាមេរិកណាដំឡើកការជាមួយ baud rate ណាមួយ ។

Syntax :

```
Serial.begin(speed) ;  
Serial.begin(speed, config)
```

Parameters:

`speed`: in bits per second (baud rate) - long

`config`: sets data, parity, and stop bits. Valid values are :

`SERIAL_5N1`

`SERIAL_6N1`

`SERIAL_7N1`

`SERIAL_8N1` (the default)

`SERIAL_5N2`

`SERIAL_6N2`

`SERIAL_7N2`

`SERIAL_8N2`

`SERIAL_5E1`

`SERIAL_7O2`

`SERIAL_8O2`

ឧចាយន័ែ៖

```
void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() { }
```

- if (Serial) : ប្រើសម្រាប់ពេលវេលាដែល serial port ត្រូវបានត្រួតពិនិត្យបាន ហើយ

ឧចាយន័ែ៖

```
void setup() {
    //Initialize serial and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB
    }
}

void loop() {
    //proceed normally
}
```

- available() :

ប្រើសម្រាប់ពេលវេលាដែលទីនេះមិនមែនមានការបញ្ចូនឯងទេ ប៉ុន្មាននឹងអាជីវកម្មនៃការបញ្ចូនឯង។
(Buffer គឺជាភ្លាមានឯងដែលបានបញ្ចូនឯងឡើងដោយ Serial Monitor អតិថិជនការណ៍)

Syntax :

Serial.available()

Arduino Mega only:

Serial1.available()

Serial2.available()

Serial3.available()

ឧត្តមន៍ ១៖

```

int incomingByte = 0; // for incoming serial data
void setup() {
    Serial.begin(9600); // opens serial port, sets data
    rate to 9600 bps
}
void loop() {
    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();
        // say what you got:
        Serial.print("I received: ");
        Serial.println(incomingByte, DEC);
    }
}

```

ឧត្តមន៍ ២ ៖ សម្រាប់ប្រើជាមួយ Arduino Mega

```

void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
}
void loop() {
    // read from port 0, send to port 1:
    if (Serial.available()) {
        int inByte = Serial.read();
        Serial1.print(inByte, BYTE);
    }
    // read from port 1, send to port 0:
    if (Serial1.available()) {

```

```

        int inByte = Serial1.read();
        Serial.print(inByte, BYTE);
    }
}

```

- **end () :** ប្រើសម្រាប់បិទប្រពិបត្តិការណ៍របស់ដើង Serial ហើយឡាតាំងដើង Digital input/Output ធ្វើតាម ។

Syntax :

Serial.end()

Arduino Mega only:

Serial1.end()

Serial2.end()

Serial3.end()

ឧត្តមាឌ៖

```

void setup() {
    Serial.end();
    pinMode ( 0, INPUT);
    pinMode ( 1, INPUT);
}

```

- **find () :** ប្រើសម្រាប់ស្វែងរកពាក្យ (string) ឱ្យមួយក្នុង Buffer ។ ប្រសិនជារកឃើញពាក្យនោះវានឹង return true ផ្សេងៗវិញ false នៅពេលដល់ time out ។

Syntax :

Serial.find(targetString)

ឧត្តមាឌ៖

```

void setup() {
    Serial.begin(9600);}
void loop (){
    if(Serial.available() > 0 ){

```

```

        if(Serial.find("OK")){
            int x = Serial.read(); }
        }
    }

```

- **find Until () :** ប្រើសម្រាប់ស្លែងរកពាក្យ (string) លាងម្នាយក្នុង Buffer រហូតដល់រកពាក្យនោះយើង បុចិនចិងទៅរហូតដល់រក terminal string ត្រូវបានរកយើង ។ ប្រសិនជាអកយើងពាក្យនោះវានឹង return true ផ្ទួយទៅវិញ false នៅពេលដល់ time out ។

Syntax :

```
Serial.findUntil(targetString, terminal)
```

ឧទាហារណ៍៖

```

void setup() {
    Serial.begin(9600);}
void loop (){
    if(Serial.available() > 0 ){
        if(Serial.findUntil (" OK ", "#")){
            int x = Serial.read(); }
    } }

```

- **print () :** ប្រើសម្រាប់បញ្ចូនទិន្នន័យពី Arduino Serial របស់វាទៅខាងក្រោមឱ្យបានរាយដីទេ ។

Syntax :

```

Serial.print(val)
Serial.print(val, format)

```

ឧទាហារណ៍៖

```

Serial.print(78); // give "78"
Serial.print(78, BIN); // give " 1001110"
Serial.print("welcome to RPITSSR"); // give " welcome to RPITSSR"

```

- **println () :** ប្រើសម្រាប់បញ្ចូនទិន្នន័យពី Arduino Serial របស់វាទៅខាងក្រោមឱ្យបានរាយដីទេ នៅលាងត្រូវបានបញ្ជាប់ទៅជាម្នាយនូវ ASCII (ASCII 10 or "\n")ចុះបញ្ជាត់ ។

Syntax :

```
Serial.println(val)
Serial.println(val, format)
```

ឧត្ថមន៍ខ្លះ

```
int analogValue = 0; // variable to hold the analog value
void setup() {
    // open the serial port at 9600 bps:
    Serial.begin(9600);
}
void loop() { // read the analog input on pin 0:
    analogValue = analogRead(0); // print it out in many formats:
    Serial.println(analogValue); // print as an ASCII-encoded decimal
    Serial.println(analogValue, DEC); // print as an ASCII-encoded decimal
    Serial.println(analogValue, HEX); // print as an ASCII-encoded
    hexadecimal
    Serial.println(analogValue, OCT); // print as an ASCII-encoded octal
    Serial.println(analogValue, BIN); // print as an ASCII-encoded binary
    delay(10); // delay 10 milliseconds before the next reading
}
```

- **read () :** ប្រើសម្រាប់ទាញយកទិន្នន័យ 1 byte ពីក្នុង Buffer ។
ទិន្នន័យដែលទទួលបានគឺជាកំណែលលេខ

Syntax :

```
Serial.read()
```

Arduino Mega only:

```
Serial1.read()
Serial2.read()
Serial3.read()
```

ឧត្តមាងនៃ៖

```

int incomingByte = 0; // for incoming serial data

void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {           // send data only when you receive data:
    if (Serial.available() > 0) { // read the incoming byte:
        incomingByte = Serial.read(); // say what you got:
        Serial.print("I received: ");
        Serial.println(incomingByte, DEC);
    }
}

```

- **readByte () :** ប្រើសម្រាប់ យកទិន្នន័យបានប្រើន byte ពីក្នុង Buffer ។ រួចដាក់ទៅក្នុង Array of byte [] or char [] ណាមួយដែលអ្នកតានា
តាមប្រើនឹងតួអក្សរដែលបើងចង់បាន ។

Syntax :

Serial.readBytes(buffer, length)

ឧត្តមាងនៃ៖

```

Char text [ 50 ];

void setup () {
    Serial.begin (9600);
}

void loop ( ) {
    if (Serial.available () > 0 ) {
        int x = Serial.readBytes(text,10);
    }
}

```

- **readBytesUntil() :** ប្រើដូច readBytes ដែលតែខ្លួនត្រូវការណាយកទិន្នន័យរហូតដល់ char Terminator ត្រូវបានរកដើរ

Syntax :

Serial.readBytesUntil(character, buffer, length)

ឧចាសេខ៊ែវ៖

```
Char text [ 50 ];
void setup ( ) {
    Serial.begin (9600);
}
void loop ( ) {
    if (Serial.available () > 0 ) {
        int x = Serial.readBytesUntil ('x',text,10);
    }
}
```

- **readString () :** ប្រើសម្រាប់អនុញ្ញត្តបំពាក់ឈានឱយទាំងអស់ក្នុង Buffer រួចបំលែងជាពាណក្ស ។

Syntax :

Serial.readString()

ឧចាសេខ៊ែវ៖

```
String text ;
void setup ( ) {
    Serial.begin (9600);
}
void loop ( ) {
    if (Serial.available () > 0 ) {
        txt = Serial.readString();
    }
}
```

- **readStringUntil () :** ប្រើសម្រាប់អនុញ្ញត្តបំពាក់ឈានឱយទាំងអស់ក្នុង Buffer រហូតដល់ String Terminator ត្រូវបានករឡើង រួចបំលែងជាពាណក្ស ដល់ time out ។

Syntax :

Serial.readStringUntil (terminator)

ឧទាហរណ៍៖

```
String text ;
void setup () {
    Serial.begin (9600);
}
void loop () {
    if (Serial.available () > 0 ) {
        txt = Serial.readStringUntil("ok");
    }
}
```

- **setTimeout () :** ប្រើសម្រាប់ការកំណត់រយៈពេល Arduino ត្រូវការធ្វើប្រពិបត្តិការអានទិន្នន័យ ប្រសិនជាកម្មភាពអាងរយៈពេលនៅទីផ្សារនៅវានឹងបាត់បង់ទិន្នន័យខ្លះ។

Syntax :

Serial.setTimeout(time)

Parameters:

time : timeout duration in milliseconds (long).

ឧទាហរណ៍៖

```
String text ;
void setup () {
    Serial.begin (9600);
    Serial.setTimeout(100)
}
void loop () {
    if (Serial.available () > 0 ) {
        txt = Serial.readStringUntil("ok");
    }
}
```

- **write () :** ប្រើសម្រាប់បញ្ជីនិន្ទន់យើ 1 byte ដែលពី Arduino ទៅក្រោម

Syntax :

```
Serial.write(val)
Serial.write(str)
Serial.write(buf, len)
```

Arduino Mega also supports: Serial1, Serial2, Serial3 (in place of Serial)

Parameters:

- val: a value to send as a single byte
- str: a string to send as a series of bytes
- buf: an array to send as a series of bytes
- len: the length of the buffer

ឧត្តមាឌ់ខ្លះ

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

void loop(){
    Serial.write(45); // send a byte with the value 45
    int bytesSent = Serial.write("hello"); //send the string "hello" and
    return the length of the string.
}
```

- **serialEvent () :** គឺជា Function ម្នយដែលរាយធ្វើការនៅពេលដែលមានិន្ទន់យច្ចូល
មកក្នុង Buffer នៃស៊ីសារ Arduino ។

Syntax :

```
void serialEvent(){
    //statements
}
```

Syntax for Arduino Mega only:

```

void serialEvent1(){
    //statements
}

void serialEvent2(){
    //statements
}

void serialEvent3(){
    //statements
}

```

៤. String Objects

String ជាបណ្តុះនៃអក្សររប្រើនចូលត្រូវដើរពាណាពក្ស ។ អក្សរនឹងមួយមានតម្លៃស្រី 1 byte ហើយវាគារពន្លោតាមតារាង ASCII code ៖

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	'
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	-	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	,	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	:	91	5B]	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	{DEL}

Arduino ផ្តល់ឡើងទៅ Function មួយចំនួនសម្រាប់ប្រើប្រាស់ជាមួយ string មានដូចជា៖

- ❖ String()
- ❖ charAt()
- ❖ compareTo()
- ❖ concat()
- ❖ c_str()
- ❖ endsWith()
- ❖ equals()
- ❖ equalsIgnoreCase()
- ❖ getBytes()
- ❖ indexOf()
- ❖ lastIndexOf()
- ❖ length()
- ❖ remove()
- ❖ replace()
- ❖ reserve()
- ❖ setCharAt()
- ❖ startsWith()
- ❖ substring()
- ❖ toCharArray()
- ❖ toInt()
- ❖ toFloat()
- ❖ toLowerCase()
- ❖ toUpperCase()
- ❖ trim()

➤ **String () :** សម្រាប់បង្កើតជា String ឬពី Data Type ផ្លូវការ

Syntax :

```
String(val)
String(val, base)
String(val, decimalPlaces)
```

Parameters:

val: a variable to format as a String - string, char, byte, int, long,
unsigned int, unsigned long, float, double
base (optional) - the base in which to format an integral value
decimalPlaces (only if val is float or double) - the desired decimal
places

ឧទាហរណ៍៖

```
String stringOne = "Hello String";           // using a constant String
String stringOne = String('a');               // converting a constant char into a String
String stringOne = String(stringTwo + " with more"); // concatenating two strings
String stringOne = String(13);                 // using a constant integer
String stringOne = String(analogRead(0), DEC); // using an int and a base
String stringOne = String(45, HEX);            // using an int and a base (hexadecimal)
String stringOne = String(255, BIN);            // using an int and a base (binary)
String stringOne = String(millis(), DEC);      // using a long and a base
String stringOne = String(5.698, 3);           // using a float and the decimal places
```

➤ **charAt () :** សម្រាប់បង្កើតចាប់យកអក្សរនៅត្រង់ Index លាងម្អួយ

Syntax :

```
string.charAt(n) ;
```

Parameters:

string: a variable of type String
n: the character to access

- **compareTo ()** : ប្រើសម្រាប់ប្រើប្រាស់ជាប្រព័ន្ធបន្ថែម String 2

Syntax : string.compareTo(string2)

Parameters :

string: a variable of type String

string2: another variable of type String

Returns:

a negative number: if string comes before string2

0: if string equals string2

a positive number: if string comes after string2

- **concat ()** : ប្រើសម្រាប់ប្រើប្រាស់ជាបញ្ហាល String គ្នា

Syntax :

string.concat(string, string2)

Parameters :

string, string2: variables of type String

Returns :

new String that is the combination of the original two Strings

ឧត្តមាវគ្គោះ

```
String stringTwo = " A long interger:";
```

```
stringTwo.concat(123456789); // " A long interger: 123456789"
```

- **endsWith ()** : ប្រើសម្រាប់តែស្ថិតិថី string បានបញ្ចប់ដោយពាក្យណាមួយវិវត្ថ

Syntax :

string.endsWith(string2)

Parameters:

string: a variable of type String

string2: another variable of type String

Returns:

true: if string ends with the characters of string2

false: otherwise

- **equals ()** : ប្រើសម្រាប់បង្កើត string 2 ថា ពីស្ថិតិយវត្ថុអត់ ហើយអនុញ្ញាត និងជំនួយ។

Syntax :

```
string.equals(string2)
```

Parameters:

string, string2: variables of type String

Returns :

true: if string equals string2

false: otherwise

- **equalsIgnoreCase ()** : ប្រើសម្រាប់បង្កើត string 2 ថា ពីស្ថិតិយវត្ថុអត់ ហើយមិនគួរពនៃអក្សរក្នុង និងជំនួយ។

Syntax:

```
string.equalsIgnoreCase(string2)
```

Parameters:

string, string2: variables of type String

Returns:

true: if string equals string2 (ignoring case)

false: otherwise

- **getBytes ()** : ប្រើសម្រាប់ចម្លង String ជាកំចុល Array លាមួយ

Syntax :

```
string.getBytes(buf, len)
```

Parameters:

string: a variable of type String

buf: the buffer to copy the characters into (byte [])

len: the size of the buffer (unsigned int)

- **indexof ()** : ប្រើសម្រាប់រក index នៃអក្សរក្នុង string ។ រកពីមុខទៅមុន

Syntax:

```
string.indexOf(val)
string.indexOf(val, from)
```

Parameters:

string: a variable of type String
val: the value to search for - char or String
from: the index to start the search from

Returns:

The index of val within the String, or -1 if not found.

- **lastIndexof ()** : ប្រើសម្រាប់រក index នៃអក្សរក្នុង string ។ រកពីក្រោយទៅមុន

Syntax :

```
string.lastIndexOf(val)
string.lastIndexOf(val, from)
```

Parameters :

string: a variable of type String
val: the value to search for - char or String
from: the index to work backwards from

Returns :

The index of val within the String, or -1 if not found.

- **length ()** : ប្រើសម្រាប់រាសចំនួនអក្សរក្នុងមួយ string

Syntax :

```
string.length()
```

Parameters :

string: a variable of type String

Returns :

The length of the String in characters.

- **remove ()** : ប្រើសម្រាប់លើបង្កូនក្នុង string ពីត្រឹម index ណាមួយរហូតដល់ទីបញ្ហា

Syntax:

```
string.remove(index)
string.remove(index, count)
```

Parameters :

index: a variable of type unsigned int
count: a variable of type unsigned int

- **replace () :** ប្រើសម្រាប់ដំឡើសពាក្យណាមួយតុង String ដោយពាក្យថ្មី

Syntax :

```
string.replace(substring1, substring)
```

Parameters :

string: a variable of type String
substring1: another variable of type String
substring2: another variable of type String

Returns :

another String containing the new string with replaced characters.

- **setCharAt () :** ប្រើសម្រាប់ដោកអក្សរមួយតុងទៅក្នុង string នៅ index ណាមួយ

Syntax :

```
string.setCharAt(index, c)
```

Parameters :

string: a variable of type String
index: the index to set the character at
c: the character to store to the given location

- **startsWith () :**

ប្រើសម្រាប់តែសមិទ្ធភាពពាក្យតុង string បានផ្តើមដោយពាក្យណាមួយប្រចាំទេរ

Syntax :

```
string.startsWith(string2)
```

Parameters:

string, string2: variable2 of type String

Returns :

true: if string starts with the characters of string2

false: otherwise

> `substring ()` :

ប្រើសម្រាប់កាត់ពាក្យពីត្រីម index លាម្អួយទៅកាន់ index លាម្អួយទៀត

Syntax :

`string.substring(from)`

`string.substring(from, to)`

Parameters :

string: a variable of type String

from: the index to start the substring at

to (optional): the index to end the substring before

> `toInt ()` :

ប្រើសម្រាប់បំលែងពាក្យ អោយទៅជាលេខ ចំនួនគត់ (int)

Syntax :

`string.toInt()`

Parameters :

string: a variable of type String

Returns :

+ long

+ If no valid conversion could be performed because the string doesn't start with a integer number, a zero is returned.

> `toFloat ()` :

ប្រើសម្រាប់បំលែងពាក្យ អោយទៅជាលេខ ចំនួនទស្សនក (Float)

Syntax :

`string.toFloat()`

Parameters :

string: a variable of type String

Returns :

- + float
- + If no valid conversion could be performed because the string doesn't start with a digit, a zero is returned.

> toLowerCase () :

ប្រើសម្រាប់បំលែងពាក្យ ដែលផ្តើបាយអក្សរជំទៅអក្សរតូចទាំងអស់

Syntax:

```
string.toLowerCase()
```

Parameters :

string: a variable of type String

> toUpperCase () :

ប្រើសម្រាប់បំលែងពាក្យ ដែលផ្តើបាយអក្សរតូចទៅជាមក្សរជំទៅអស់

Syntax:

```
string.toUpperCase()
```

Parameters :

string: a variable of type String

> trim () :

ប្រើសម្រាប់កាត់ចេញនូវ whitespace ពីមុខ និងពីក្រាយពាក្យបស់យើង

Syntax:

```
string.trim()
```

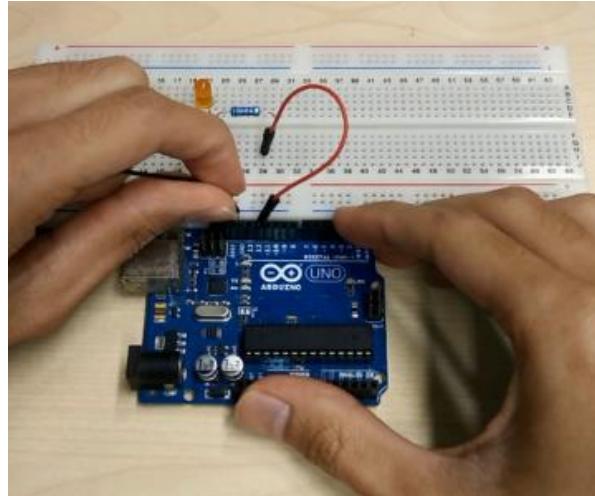
Parameters :

string: a variable of type String

គំណើន ៥ : Arduino Projects

1. ពិសោធន៍ ១ : Digital Output
2. ពិសោធន៍ ២ : LED Blinking
3. ពិសោធន៍ ៣ : Light Chaser
4. ពិសោធន៍ ៤ : Digital Input
5. ពិសោធន៍ ៥ : Tilt Sensor
6. ពិសោធន៍ ៦ : Single Digit 7-segment
7. ពិសោធន៍ ៧ : 2 Digit 7-segment make counter 00-99
8. ពិសោធន៍ ៨ : Control LED using Serial Command
9. ពិសោធន៍ ៩ : Temperature Sensor (LM35)
10. ពិសោធន៍ ១០ : Light Dependent Resistor
11. ពិសោធន៍ ១១ : LCD display
12. ពិសោធន៍ ១២ : IR Remote
13. ពិសោធន៍ ១៣ : Servo Motor
14. ពិសោធន៍ ១៤ : DC Motor
15. ពិសោធន៍ ១៥ : Step Motor
16. ពិសោធន៍ ១៦ : Distance Sensor
17. ពិសោធន៍ ១៧ : Gas Sensor
18. ពិសោធន៍ ១៨ : KeyPad
19. ពិសោធន៍ ១៩ : Stepper Motor
20. ពិសោធន៍ ២០ : DHT11 Senso
21. ពិសោធន៍ ២១ : Alcohol (MQ-3) sensor ប្រើជាមួយ LCD display និង Buzzer
22. ពិសោធន៍ ២២ : HC-05 Bluetooth Module

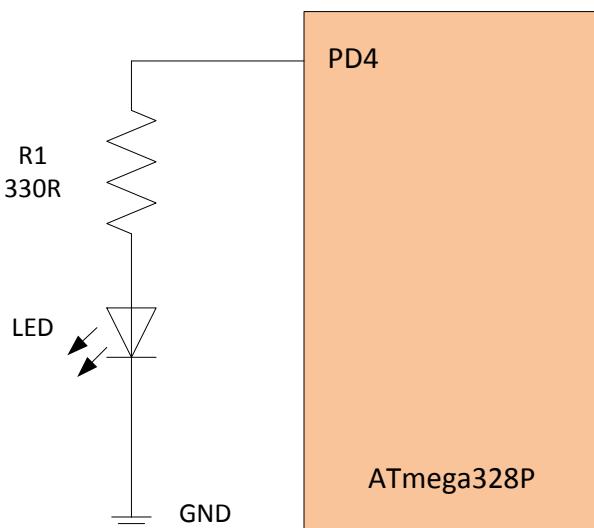
ចិត្តផល ១ : Digital Output



ការធ្វើសិសយកអំពុល LED (យុតិពន្លឹ) មកអនុវត្តន៍ទៅស្ថីដើម្បីក្រោកចេញងាយស្រួលក្នុងការតែស្ថិនិងការត្រួតប័រហេតុ ហើយប្រសិនបើវត្ថុ (HIGH) វាកំពងដំណើរការ និងលបតា (LOW) វាមិនដំណើរការដែលអ្នកបញ្ជាញព័មានពីដើរបស់ Microcontroller ។ ជាទុទៅមានការត្រួតប័ស្ថីពីរប្រភេទ សម្រាប់ប្រកចេញ (OUTPUT) ពីដើរបស់ Microcontroller ។

- ❖ ការយកចរន្តចេញពី Microcontroller ហើយថា Current sourcing
- ❖ ការអាយចរន្តចូលពី Microcontroller ហើយថា Current sinking ។

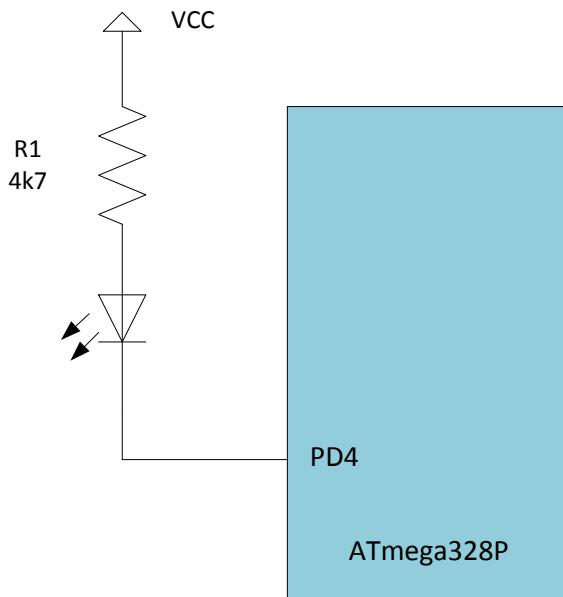
១. សេវ្យតិតិ Current Sourcing



លទ្ធផល: ជំនើយខាងក្រោម

- អំពុល LED ត្រួត ដើរ PD4 តី HIGH (1)
- អំពុល LED រលក់ ដើរ PD4 តី LOW (0)

២. នេត្តក្លឹង Current sinking



លទ្ធផល: ជំនើយខាងក្រោម

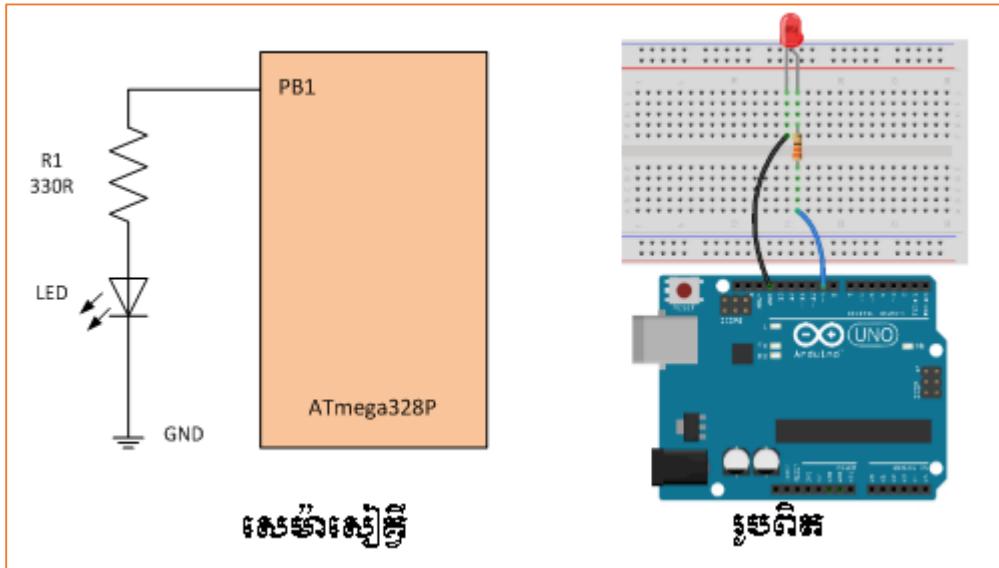
- អំពុល LED ត្រួត ដើរ PD4 តី LOW (0)
- អំពុល LED រលក់ ដើរ PD4 តី HIGH (1)

សរស់រក្សាទុកដោយដើរដើរ និង របស់ Arduino UNO នៅលើ LED ត្រួត (ជំនើយការ) រហូតដូច្នៃបាន ខាងក្រោម:

ក្រុមការងារ

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. 330Ω resistor – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED – 1 គ្រាប់

គននេនុទត្តិនត្តិ



អ្នក :

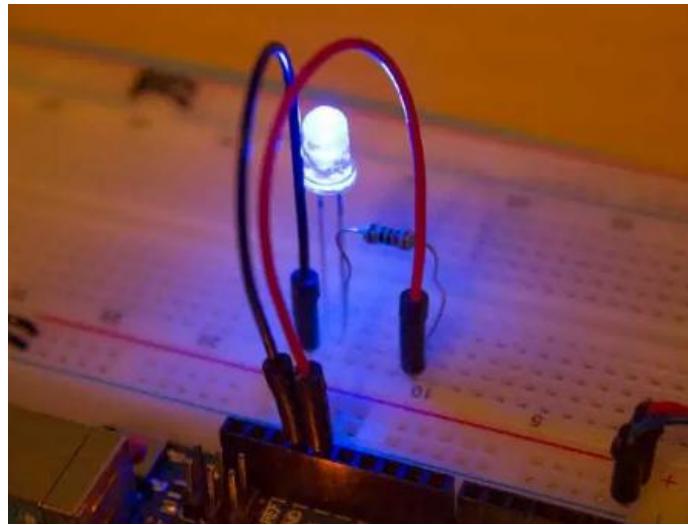
```
void setup() {
    // put your setup code here, to run once:
    pinMode(9,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(9,HIGH);
}
```

លំហាត់អនុទត្តិ

- សរស់រក្សាទិន៍ដើម្បី របស់ Arduino UNO អាយ LED ភ្លើ (ដំណើរការ) រហូត
- សរស់រក្សាទិន៍ដើម្បី របស់ Arduino UNO អាយ LED 2 គ្រប់ភ្លើ (ដំណើរការ)
- សរស់រក្សាទិន៍ របស់ Arduino UNO អាយ LED ភ្លើ (ដំណើរការ) ទាំងអស់ទាំង 14 ដើម្បី

តិចលោខណ៍ ២ ៖ LED Blinking



Delay Time (ការពន្លាគេណៈ)

Delay time គឺជាការពន្លាគរយៈពេលដំណើរការនៃក្នុង ។ នៅក្នុង Arduino IDE , delay ថែកថ្វាតាតីរ Syntax គឺ

`delay (ms);`

`delayMicroseconds(us);`

ដែល ms: រួយៗពេលគិតជាមីលីវិនាទី (1000 មីលីវិនាទី = 1 វិនាទី)

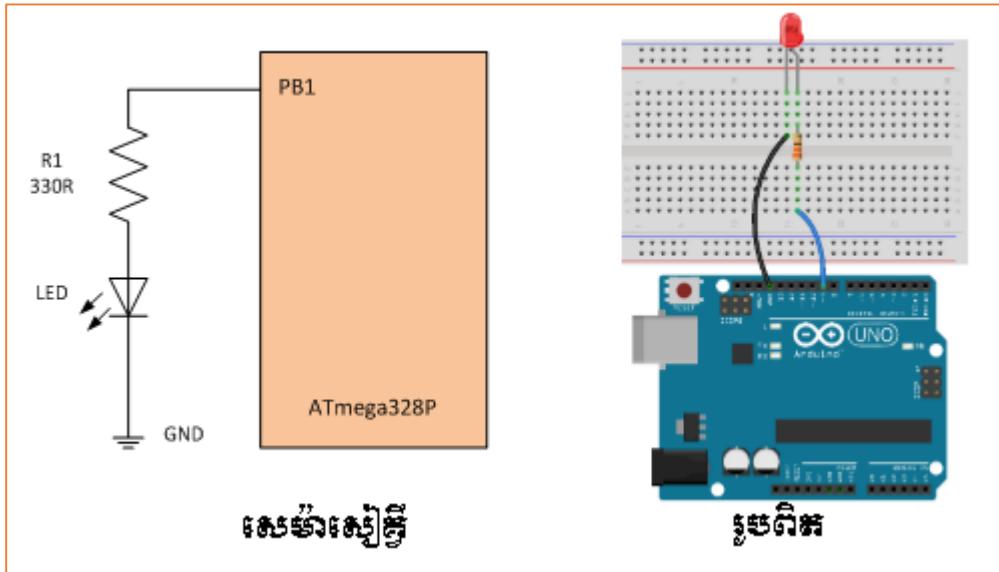
ឬ: រួយៗពេលគិតជាមីក្រុវិនាទី (1000 មីក្រុវិនាទី = 1មីលីវិនាទី)

សរស់រក្សាទុក្រុមដែលធ្វើឡើងនៅលើ Arduino UNO នាំយក LED ភ្លើលត់(Blinking) រហូតដូចមួយ ឧងត្រាយ:

ក្រុមប្រឈមបញ្ជាផលនៃបណ្តុះបណ្តាល

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. 330Ω resistor – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED – 1 គ្រាប់

គមនេនុពលន៍តាមរបៀប



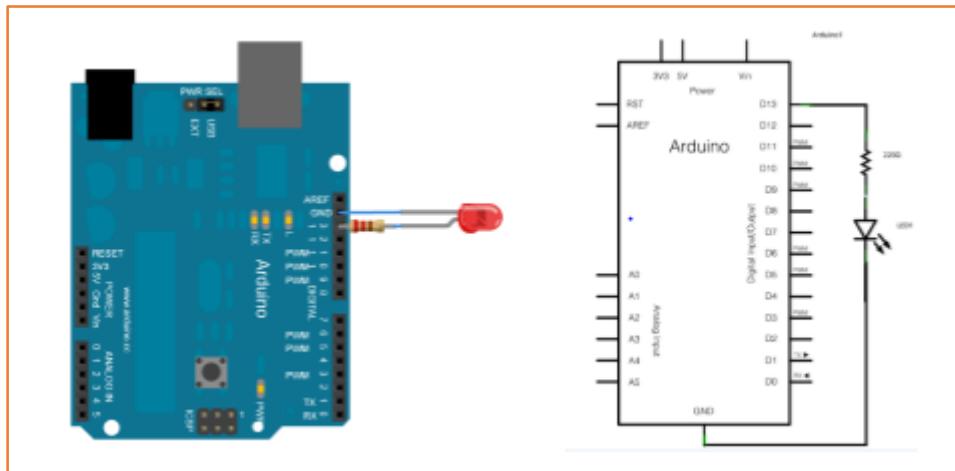
គូដា :

```
void setup() { pinMode(9,OUTPUT); }

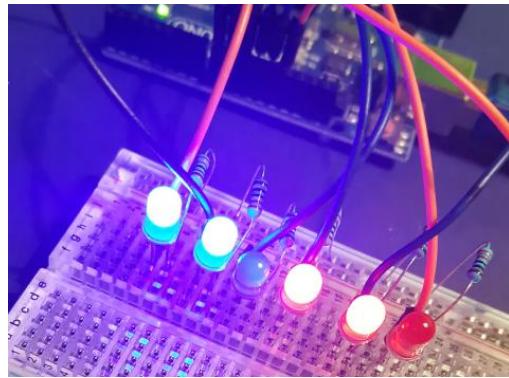
void loop() {
    digitalWrite(9,HIGH);
    delay(200);
    digitalWrite(9,LOW);
    delay(200); }
```

លំហាត់អនុពលន៍

សរស់រក្សាទិន្នន័យដើម្បី 13 របស់ Arduino UNO អាយ LED ត្រូវលាត់(Blinking) រហូតដូចរូប



តិនោជន៍ ៣ : Light Chaser

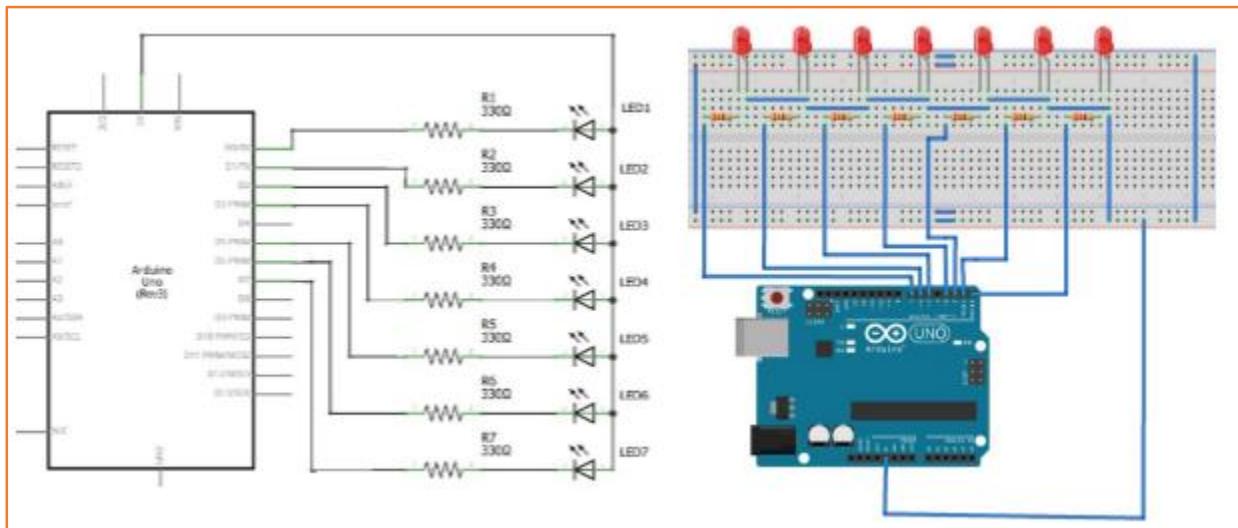


សរស់រក្សាទុដអោយធើឃើនទី 0 ដល់ធើឃើនទី 7 នៃស៊ីរីលត់ (Blinking) តាមទំនំធើឃើន។របស់អ្នកចង់បាន (លក្ខណៈនេះគេហែថា light chaser)

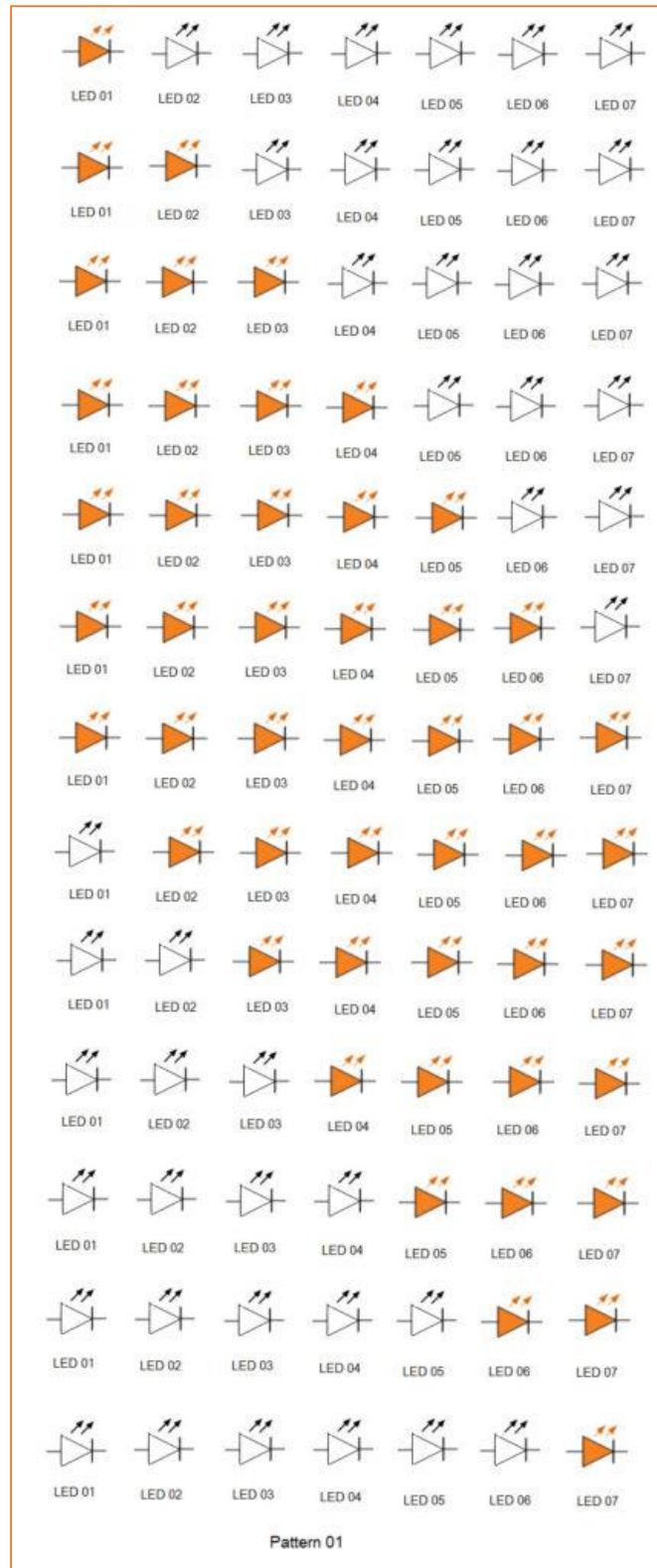
ផ្តល់នូវការងារ

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. 330Ω resistor – 7 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED – 7 គ្រាប់

រាយការណ៍ន័រ



លក្ខណៈតើនេត់ខែស៊ីតិ៍ និងការត្រួតពិនិត្យការងារ

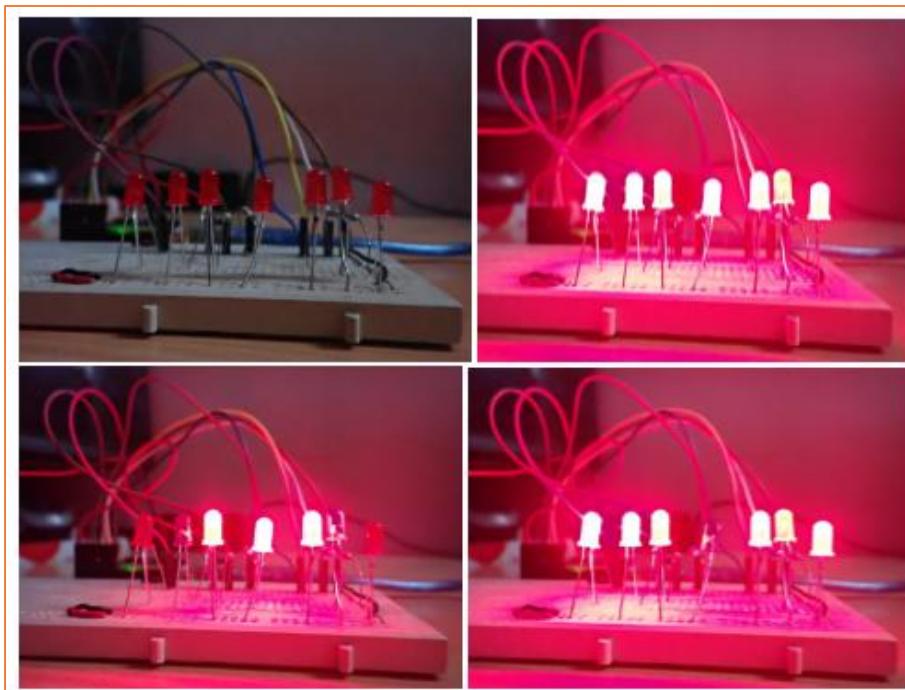


តម្លៃ :

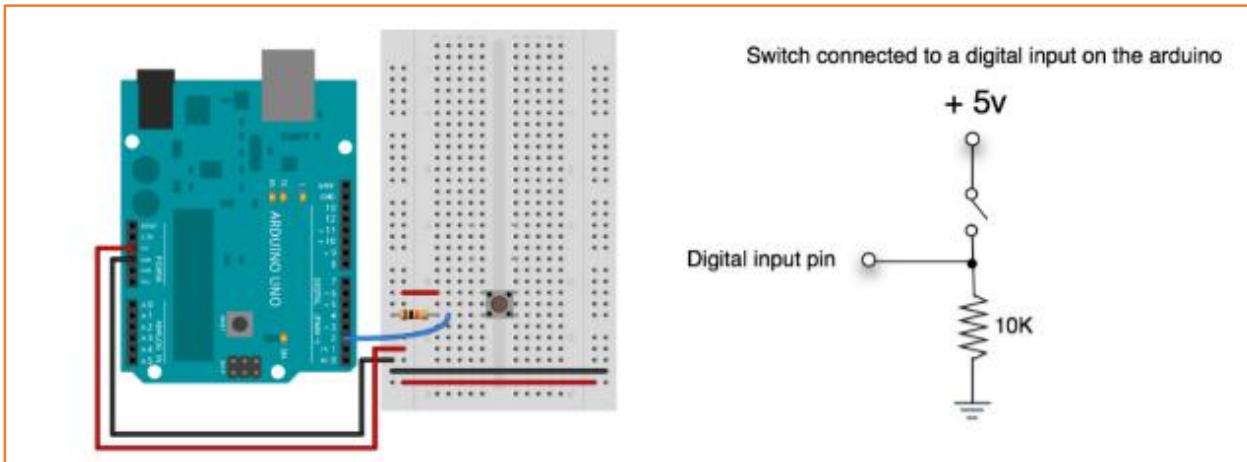
```
void setup() {  
    pinMode(0, OUTPUT);  
    pinMode(1, OUTPUT);  
    pinMode(2, OUTPUT);  
    pinMode(3, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
}  
  
void loop() { //Pattern 01  
    digitalWrite(0, LOW);  
    delay(300);  
    digitalWrite(1, LOW);  
    delay(300);  
    digitalWrite(2, LOW);  
    delay(300);  
    digitalWrite(3, LOW);  
    delay(300);  
    digitalWrite(5, LOW);  
    delay(300);  
    digitalWrite(6, LOW);  
    delay(300);  
    digitalWrite(7, LOW);  
    delay(300);  
  
    digitalWrite(0, HIGH);  
    delay(200);
```

```
digitalWrite(1, HIGH);
delay(200);
digitalWrite(2, HIGH);
delay(200);
digitalWrite(3, HIGH);
delay(200);
digitalWrite(5, HIGH);
delay(200);
digitalWrite(6, HIGH);
delay(200);
digitalWrite(7, HIGH);
delay(200);
```

លទ្ធផល :



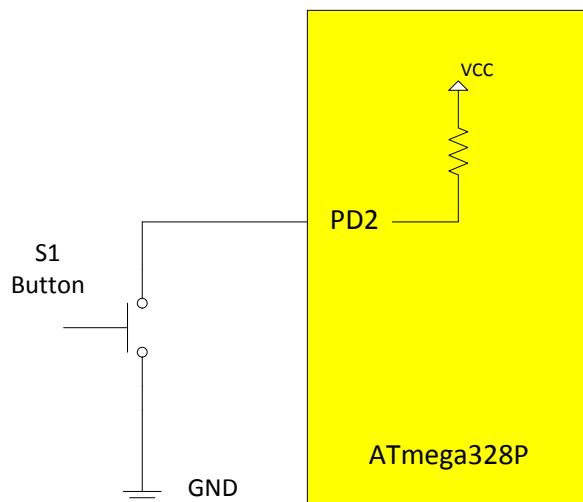
លិខោជន់ ៥: Digital Input



ការរើសយកកុងតាក់ (Switch) ឬ បូតុង (Button) មកអនុវត្តន៍ទេស្ថីស្ថី ដើម្បីប្រព័ន្ធបុលឌាយស្រួលកុងការទេស្ថីនិងការតភ្តាប់រហូតដល់មានការតភ្តាប់ស្ថីពីរប្រភេទសម្រាប់ប្រកចូល (INPUT) ដើម្បីរបស់ Microcontroller ។

- ❖ Pull-up resistor ដែកបែងបានពីគីឡូ និង External pull-up resistor
- ❖ និង Pull-down resistor ។

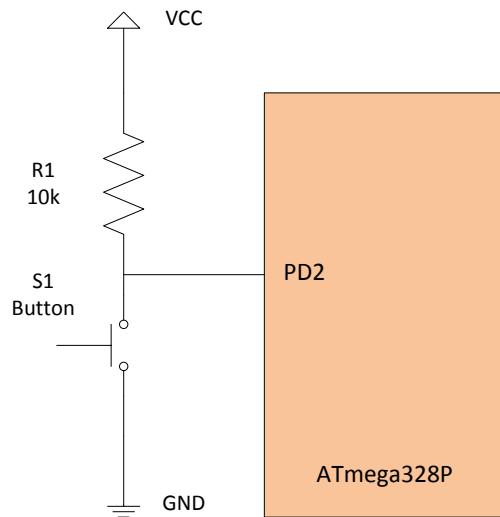
១. ឈ្មោះ Internal pull-up resistor



ធម្មនោះដឹងទៀត

- កុងតាក់ S1 តីចំហេ (open) ធ្វើឱ្យ PD2 តី HIGHT (1)
- កុងតាក់ S1 តីបិទ (close) ធ្វើឱ្យ PD2 តី LOW (0)

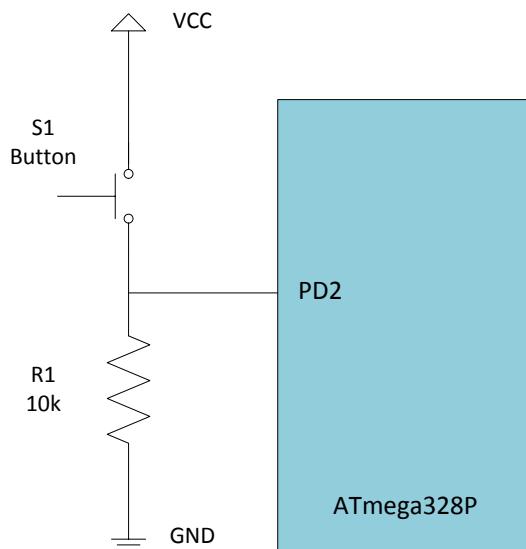
២. លេខ្លឹម External pull-up resistor



ធម្មនោះជំនួយនៅ ឲ្យមាន Internal pull-up resistor ដែល

- ក្នុងតាក់ S1 តីចិប់ (open) ធ្វើឱ្យ PD2 តី HIGHT (1)
- ក្នុងតាក់ S1 តីបិទ (close) ធ្វើឱ្យ PD2 តី LOW (0)

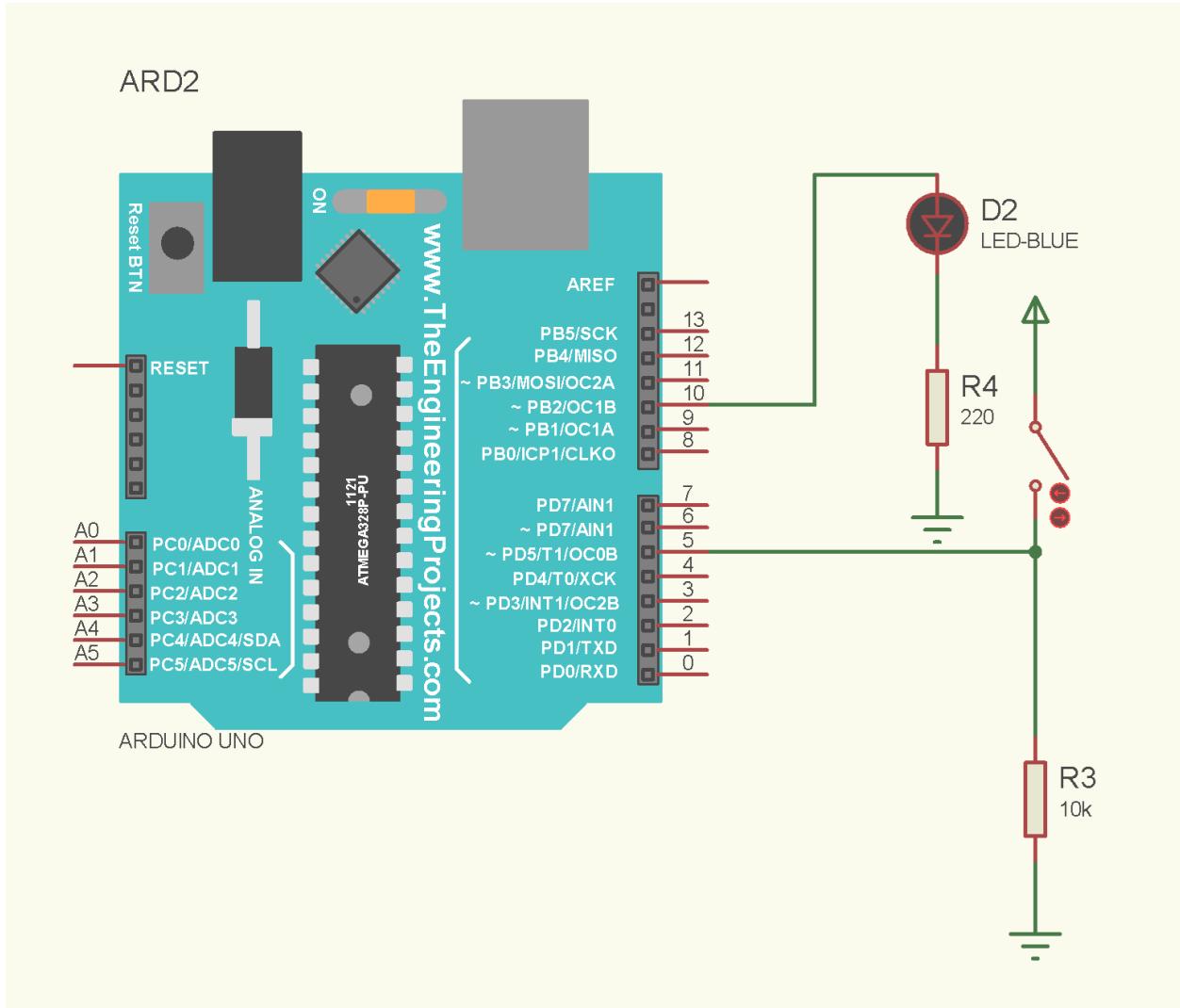
៣. លេខ្លឹម Pull-down resistor



ធម្មនោះជំនួយនៅ

- ក្នុងតាក់ S1 តីចិប់ (open) ធ្វើឱ្យ PD2 តី LOW (0)
- ក្នុងតាក់ S1 តីបិទ (close) ធ្វើឱ្យ PD2 តី HIGH (1)

អនុវត្តន៍សៀវភៅតាមយកដាក់ លក្ខណៈ: ត្រូវបែងជា Pull-down resistor
សរស់រកដោយដើរឡើងទី 10 របស់ Arduino UNO ដោយ LED ត្រូវរាលើចុចបូតុងនៅដើរឡើងទី 5
របស់ Arduino UNO ហើយអត្ថបុច្ចិតុងកើត LED មិនអាចដោះ ត្រូវបែងសៀវភៅជូនប្រាប់រក្សាម៖



ផ្តល់ :

```

int sw=5;
int led= 10;
void setup() {
    pinMode(sw ,INPUT);
    pinMode(led , OUTPUT);
}

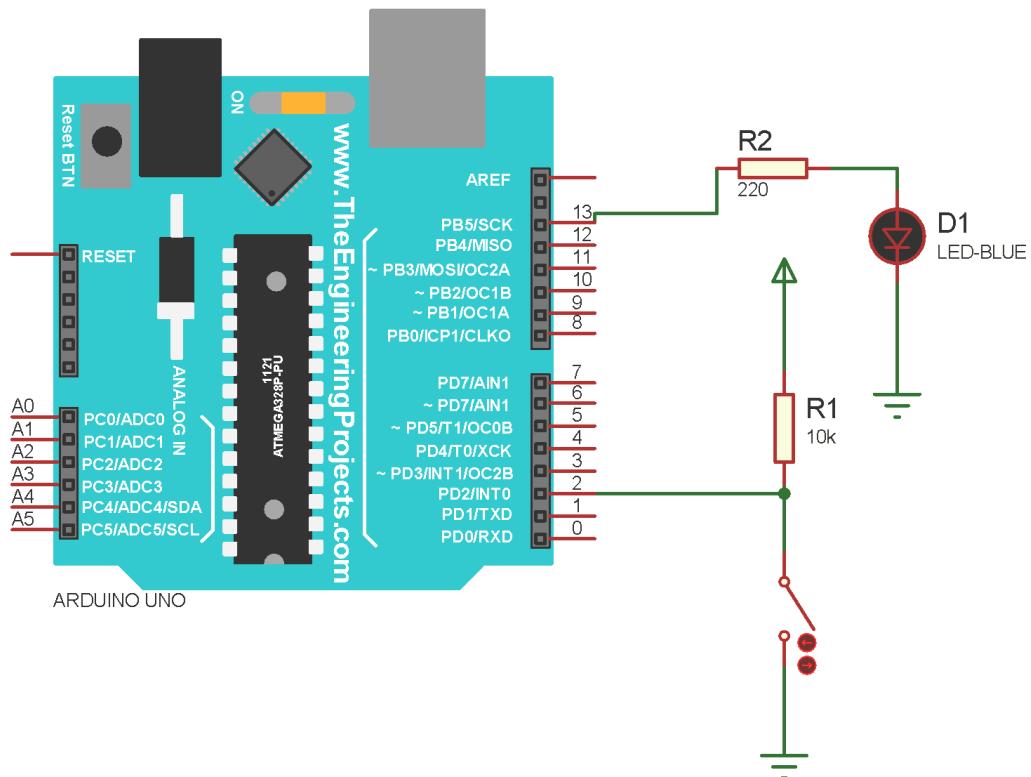
```

```

void loop() {
    if ( digitalRead ( sw ) == HIGH)
    {
        digitalWrite ( led ,HIGH);
    }
    if ( digitalRead (sw) ==LOW )
    {
        digitalWrite ( led , LOW);
    }
}

```

អនុវត្តន៍សៀវភៅដាម្បួយកុងតាក់ លក្ខណៈត្រូវបាន Pull –up resistor



លក្ខណៈ: ដំណឹងការនៅពេលចូចកុងតាក់ នាំអោយ LED1 តី អត់ចូចកុងតាក់ LED1 តីរលត់

ក្នុង :

```
int sw1=2;
int led1= 13;
void setup() {
    pinMode(sw1 ,INPUT);
    pinMode(led1 , OUTPUT); }

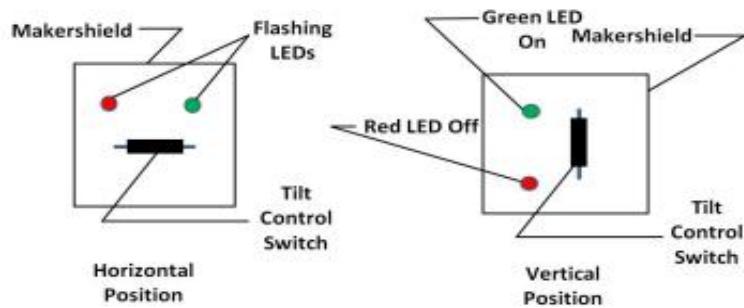
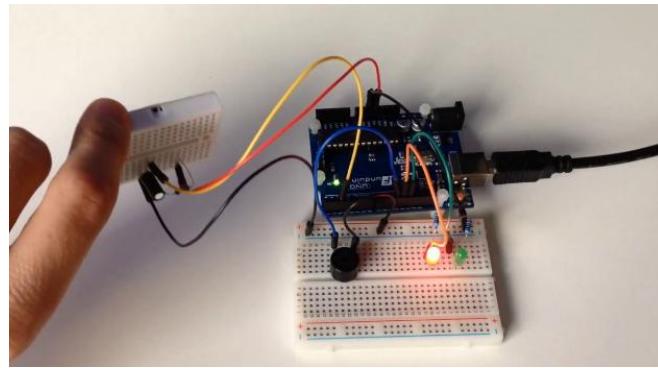
void loop() {
    if ( digitalRead ( sw1 ) == HIGH) {
        digitalWrite ( led1 ,LOW);    }
    if ( digitalRead (sw1 ) ==LOW ){
        digitalWrite ( led1 , HIGH);    }}
```

ប្រអាជសរស់រម្យភាពខ្សោត

```
int sw1=2;
int led1= 13;
void setup() {
    pinMode(sw1 ,INPUT);
    pinMode(led1 , OUTPUT);}

void loop() {
    if ( digitalRead ( sw1 ) == HIGH) {
        digitalWrite ( led1 ,LOW);    }
    else{
        digitalWrite ( led1 , HIGH);
    } }
```

តិះសែល់ ធន់ Tilt Sensor

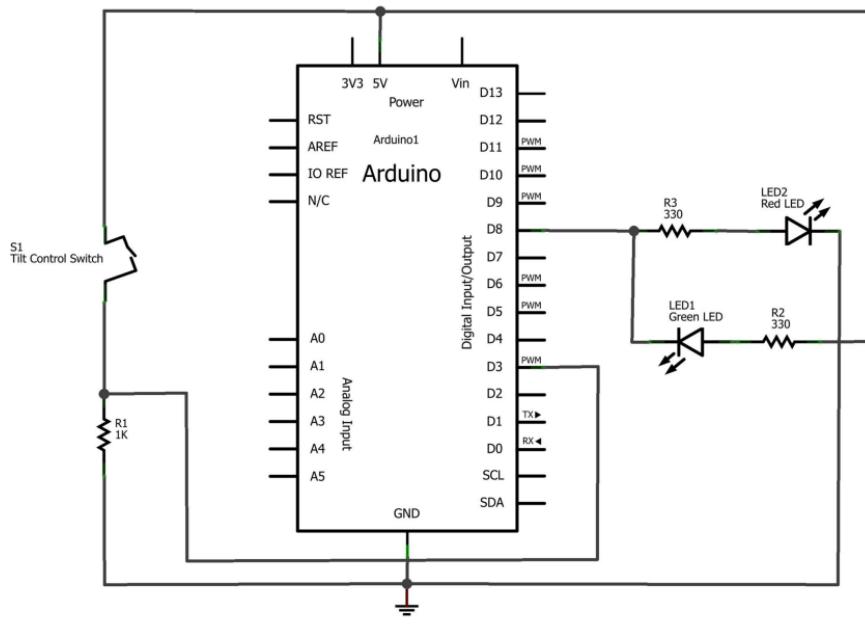


ការពិសោធន៍ជីថេតែសទ្វេ Tilt sensor (ប្រភេទ sw-520D) វាបាននឹងការផ្តើមកុងតាក់ដែរ ក្រណី ផ្លូវ អំពួលLED ពណ៌ក្រហមត្រូវតែណាមខ្សោយរលក់ ក្រណីបញ្ជី អំពួលLED ពណ៌ក្រហមរលក់តែណាមខ្សោយរក្សី សៀវភៅត្រូវបានក្រឡាម៉ាំងក្រឡាម៉ាំង។

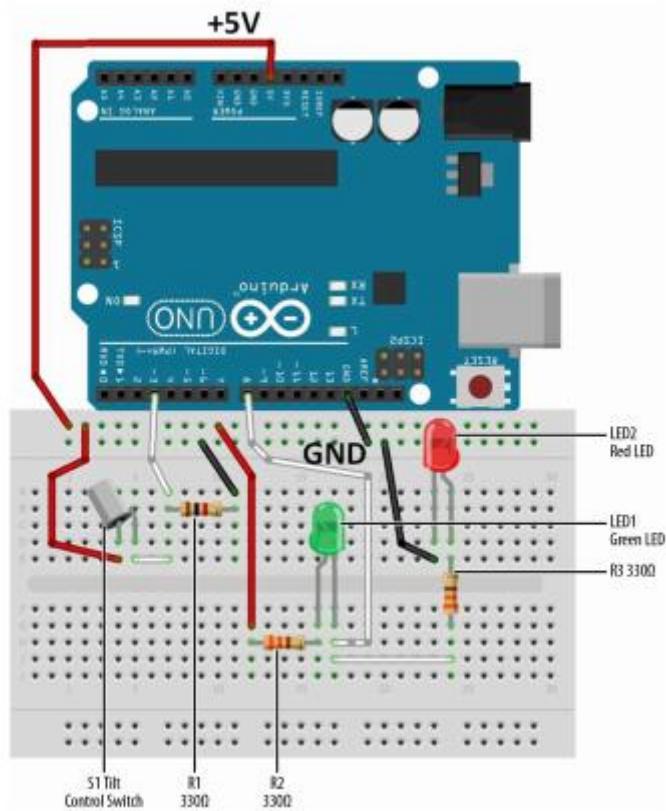
ក្រុមឈាមនូវតាមលេខាធិការ

1. Arduino Uno R3 with ATmega328P
2. 330Ω resistor – ៣ គ្រាប់
3. Jumper wires
4. Breadboard
5. LED ពណ៌ខ្សោយ - 1 គ្រាប់
6. LED ពណ៌ក្រហម – 1 គ្រាប់
7. Tilt sensor (ប្រភេទ sw-520D) – 1 គ្រាប់

គាមរុទត្ថនភាព



សេចក្តីផ្តើម

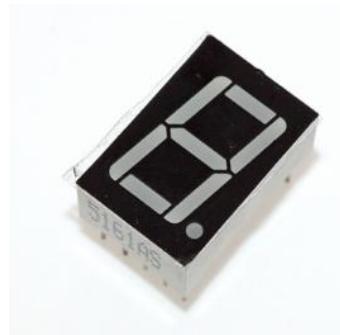


រូបពិត

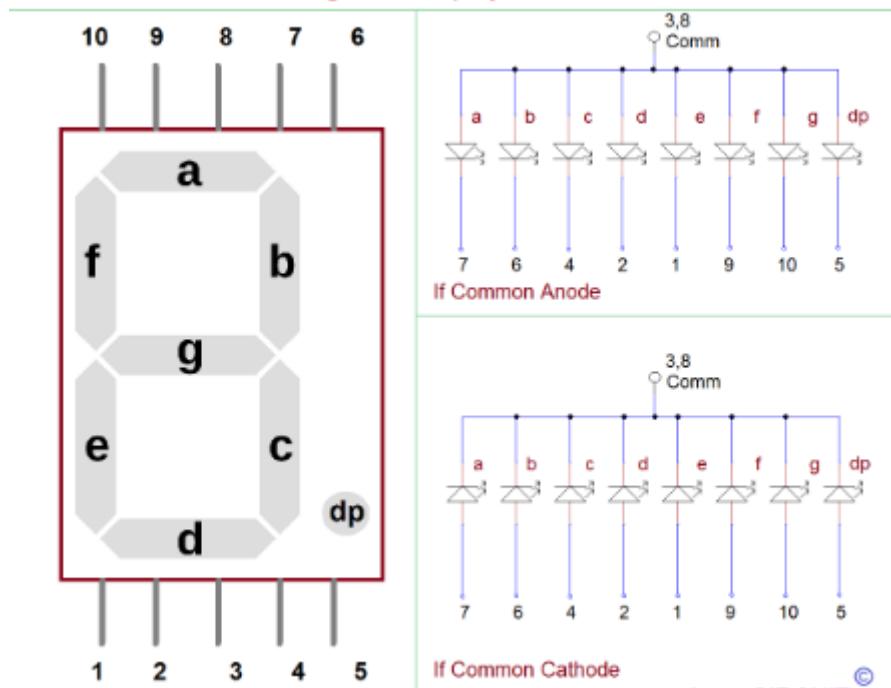
ខ្លួន :

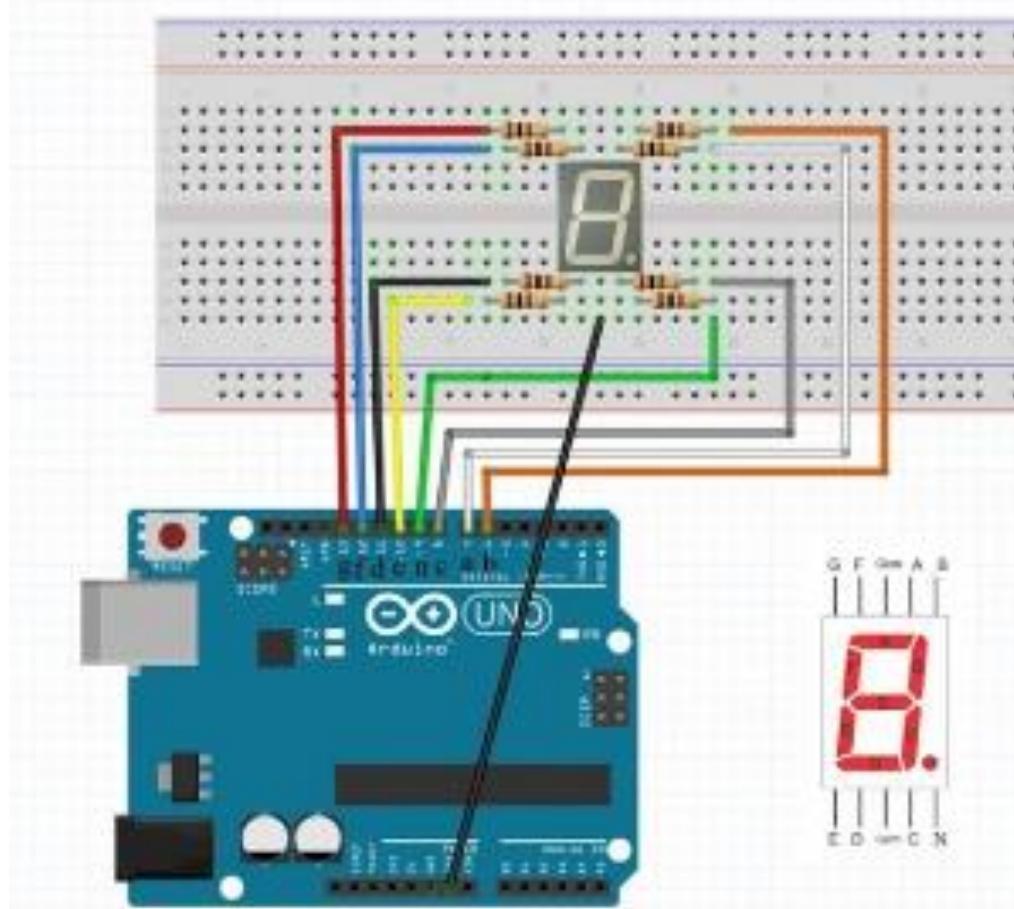
```
const int tiltPin = 3; // the number of the tilt control switch pin  
const int ledPin = 8; // the number of the LED pin  
  
// variables will change:  
  
int tiltState = 0; // variable for tilt control switch status  
  
void setup() {  
    pinMode(ledPin, OUTPUT);  
  
    // initialize the tilt control switch pin as an input:  
    pinMode(tiltPin, INPUT);  
}  
  
void loop(){ // read the state of the tilt control switch value:  
    tiltState = digitalRead(tiltPin);  
  
    // check if the tilt control switch contacts are closed  
    // if they are, the tiltState is HIGH:  
    if (tiltState == HIGH) { // turn Red LED on;  
        digitalWrite(ledPin, HIGH);  
        delay(100); // wait 100ms:  
  
        // turn LED off:  
        digitalWrite(ledPin,LOW);  
        delay(100); // wait 100ms:  
    }  
    else {  
        digitalWrite(ledPin, HIGH); // turn LED off:  
    }  
}
```

តិះនោជន់ ៦៖ Single Digit 7-segment



7 Segment Display Pinout





គ្មាន់ : បង្កាញលេខ 0 , 1, 2, 3 និងលេខ 8 ដំណើរការដែលដោរ

```
void setup() {
    pinMode(13, OUTPUT); // pin g
    pinMode(12, OUTPUT); // pin f
    pinMode(11, OUTPUT); // pin d
    pinMode(10, OUTPUT); // pin e
    pinMode(9, OUTPUT); // pin n
    pinMode(8, OUTPUT); // pin c
    pinMode(7, OUTPUT); // pin a
    pinMode(6, OUTPUT); // pin b
}
```

```
void loop() {  
    // number 0  
    digitalWrite(13, LOW);  
    digitalWrite(12, HIGH);  
    digitalWrite(11, HIGH);  
    digitalWrite(10, HIGH);  
    digitalWrite(9, HIGH);  
    digitalWrite(8, HIGH);  
    digitalWrite(7, HIGH);  
    digitalWrite(6, HIGH);  
    delay(500);  
  
    // number 1  
    digitalWrite(13, LOW);  
    digitalWrite(12, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(9, HIGH);  
    digitalWrite(8, HIGH);  
    digitalWrite(7, LOW);  
    digitalWrite(6, HIGH);  
    delay(500);  
  
    // number 2  
    digitalWrite(13, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(11, HIGH);  
    digitalWrite(10, HIGH);  
    digitalWrite(9, HIGH);  
    digitalWrite(8, LOW);
```

```
digitalWrite(7, HIGH);
digitalWrite(6, HIGH);
delay(500);

// number 3

digitalWrite(13, HIGH);
digitalWrite(12, LOW);
digitalWrite(11, HIGH);
digitalWrite(10, LOW);
digitalWrite(9, HIGH);
digitalWrite(8, HIGH);
digitalWrite(7, HIGH);
digitalWrite(6, HIGH);
delay(500);

// number 8

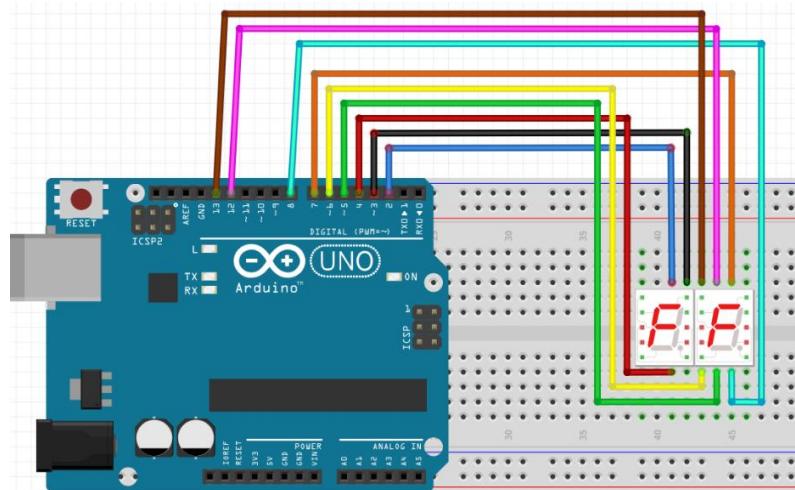
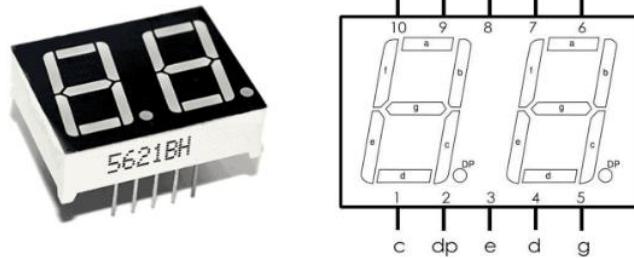
digitalWrite(13, HIGH);
digitalWrite(12, HIGH);
digitalWrite(11, HIGH);
digitalWrite(10, HIGH);
digitalWrite(9, HIGH);
digitalWrite(8, HIGH);
digitalWrite(7, HIGH);
digitalWrite(6, HIGH);
delay(500);

}
```

លំហាត់អនុវត្តន៍

ចូរសរស់រកុដបង្ហាញពីលេខ 9 មក 0 វិញ្ញានវិញ្ញាន 7segment ដូចពិសោធន៍ទី៦

តិចែលចន់ ៧៖ 2 Digit 7-segment make counter ០០-៩៩



ចូល:

ក្នុងបង្អាញពីលេខ ០០ ទៅ ៩៩ ឡើងម្បយដង ១លេខរយៈពេល ១វិនាទី

```
int digit[10] = {0b0111111, 0b0000110, 0b1011011, 0b1001111, 0b1100110,
0b1101101, 0b1111101, 0b0000111, 0b1111111, 0b1101111};

int digit1, digit2;

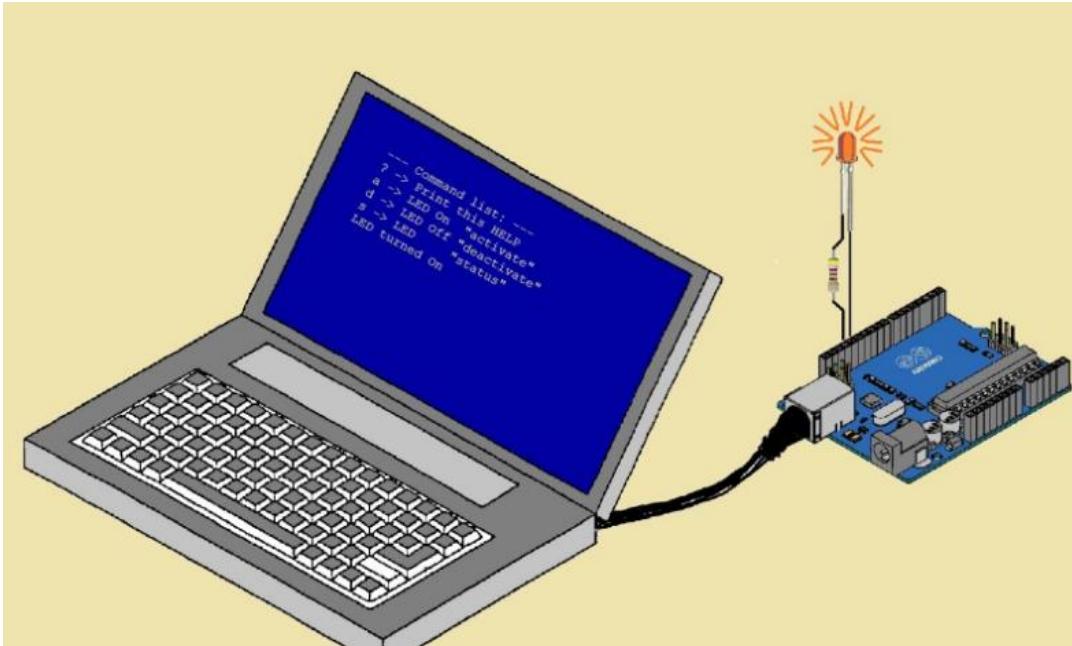
void setup() {
    for (int i = 2; i < 9; i++) {
        pinMode(i, OUTPUT);
    }
    pinMode(12, OUTPUT);
```

```
pinMode(13, OUTPUT);
}

void loop() {
    for (int j = 0; j <= 99; j++) {
        digit2 = j / 10;
        digit1 = j % 10;
        for ( int k = 0; k < 20; k++) {
            digitalWrite(12, HIGH);
            digitalWrite(13, LOW);
            dis(digit2);
            delay(10);
            digitalWrite(13, HIGH);
            digitalWrite(12, LOW);
            dis(digit1);
            delay(100);
        }
    }
}

void dis(int num) {
    for (int i = 2; i < 9; i++) {
        digitalWrite(i, bitRead(digit[num], i - 2));
    }
}
```

ពិនេរាងល់ ៤៖ Control LED using Serial Command



ពិសោធន៍ទី៤ គឺគោលបំណងបញ្ហរអំពុល LED មួយគ្រាប់ដែលភ្លាប់ដើរលើលីហីដី 13 របស់ Arduino រាយការណ៍នឹងរលកតែតាមរយៈKeyboard របស់កំពុងទៅ ហើយរាយការណ៍នឹងយោតាមរយៈ Serial Monitor ខាងក្រោមគឺជាតារកង់ Command សម្រាម៉ែបញ្ហរអំពុល LED

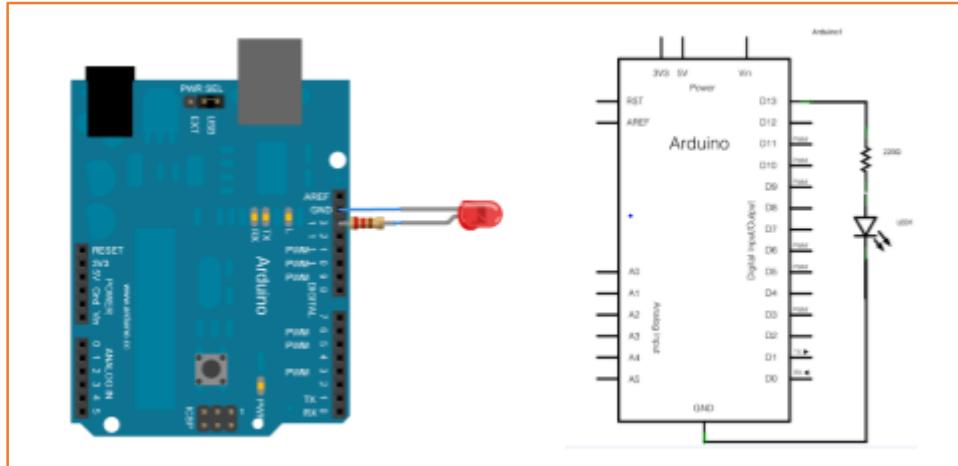
— Command list: —

- ? -> Print this HELP
- a -> LED On "activate"
- d -> LED Off "deactivate"
- s -> LED "status"

ផ្តល់នូវបញ្ជីរបាយការណ៍

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. 330Ω resistor – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED – 1 គ្រាប់

ភាសាអិនុទត្តនត្រាយ



៤

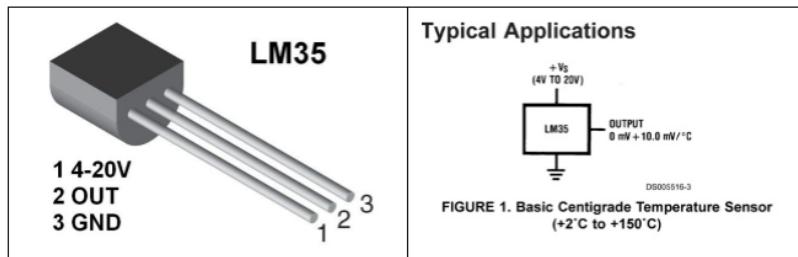
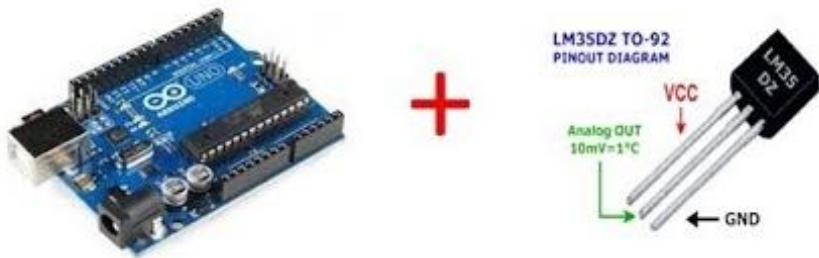
commade §

```
1 /*
2 Controlling the LED in pin 13 with the Serial Monitor.
3 --- Command list: ---
4 ? -> Print this HELP
5 a -> LED On "activate"
6 d -> LED Off "deactivate"
7 s -> LED      "status"
8 Example using the switch statement.
9 */
10
11 #define LED 13           // Pin 13 is connected to the LED
12 char rxChar= 0;         // RXcHAR holds the received command.
13
14 //== function to print the command list: =====
15 void printHelp(void){
16     Serial.println("--- Command list: ---");
17     Serial.println("? -> Print this HELP");
18     Serial.println("a -> LED On \"activate\"");
19     Serial.println("d -> LED Off \"deactivate\"");
20     Serial.println("s -> LED      \"status\"");
21 }
```

```
22
23 //----- setup -----
24 void setup() {
25     Serial.begin(9600);    // Open serial port (9600 bauds).
26     pinMode(LED, OUTPUT); // Sets pin 13 as OUTPUT.
27     Serial.flush();        // Clear receive buffer.
28     printHelp();           // Print the command list.
29 }
30
31 //----- loop -----
32 void loop() {
33     if (Serial.available() >0) {          // Check receive buffer.
34         rxChar = Serial.read();           // Save character received.
35         Serial.flush();                  // Clear receive buffer.
36
37     switch (rxChar) {
38
39         case 'a':
40         case 'A':                      // If received 'a' or 'A':
41             if (digitalRead(LED) == LOW) { // If LED is Off:
42                 digitalWrite(LED,HIGH); // Turn On the LED.
43                 Serial.println("LED turned On");
44     }
```

```
45         else Serial.println("LED already On!");
46         break;
47     case 'd':
48     case 'D':                                // If received 'd' or 'D':
49     if (digitalRead(LED) == HIGH) {           // If LED is On:
50         digitalWrite(LED,LOW);             // Turn Off the LED.
51         Serial.println("LED turned Off");
52     }
53         else Serial.println("LED already Off!"); break;
54     case 's':
55     case 'S':                                // If received 's' or 'S':
56     if (digitalRead(LED) == HIGH)           // Read LED status.
57         Serial.println("LED status: On");
58         else Serial.println("LED status: Off"); break;
59     case '?':                                // If received a ?:
60         printHelp();                      // print the command list.
61         break; | default:
62     Serial.print("'");
63     Serial.print((char)rxChar);
64     Serial.println("' is not a command!");
65   }
66 }
67 }// End of the Sketch.
```

ពិសោធន៍យ៉ាវ ៩ : Temperature Sensor (LM35)

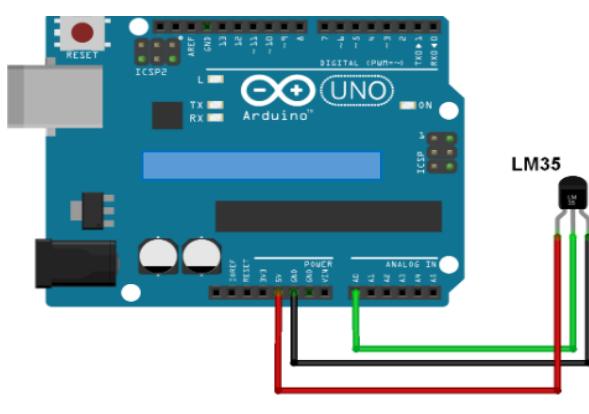


ពិសោធន៍យ៉ាវ ៩ តើគោលបំណងនៃ Temperature sensor ប្រពេទ LM35 ហើយវាបង្ហាញ
ទិន្នន័យតាមរយៈ Serial Monitor

ក្រឡើតសម្រាប់អនុវត្តន៍មែន

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. LM35 – 1 គ្រាប់
4. Jumper wires

ភាពអនុវត្តន៍ត្រួតត្រូវ



ខ្លួន:

```
const int lm35_pin = A1; /* LM35 O/P pin */  
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    int temp_adc_val;  
    float temp_val;  
    temp_adc_val = analogRead(lm35_pin); /* Read Temperature */  
    temp_val = (temp_adc_val * 4.88); /* Convert adc value to equivalent voltage */  
    temp_val = (temp_val/10); /* LM35 gives output of 10mv/°C */  
    Serial.print("Temperature = ");  
    Serial.print(temp_val);  
    Serial.print(" Degree Celsius\n");  
    delay(1000);  
}
```

ពិសោធន៍ទី 10 គិតស្ថាយស្ថាយ LDR

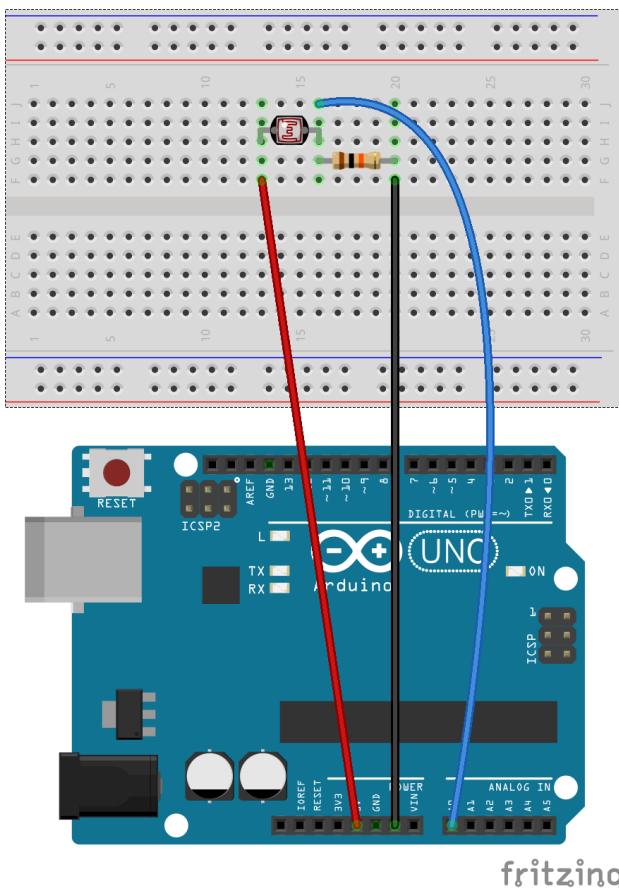
ពិសោធន៍ទី 10 គិតស្ថាយស្ថាយ LDR (light dependent resistor) ហើយរាបង្វាល់

ទិន្នន័យតាមរយៈ Serial Monitor

១. តម្រូវការស្ថើសង្គមពិសោធន៍

- ✓ Arduino Uno ១គ្រប់
- ✓ USB Cable ១ខ្សោយ
- ✓ LDR (light dependent resistor) ១គ្រប់
- ✓ ធម្មិតស្ថាយ 10 kΩ ១គ្រប់
- ✓ Breadboard
- ✓ ខ្សោយសំរាប់ត្រូវ (Jumper wire) 3 ខ្សោយ

២. ការត្រួតពិនិត្យស្ថើសង្គមពិសោធន៍



៣. ទូទៅ

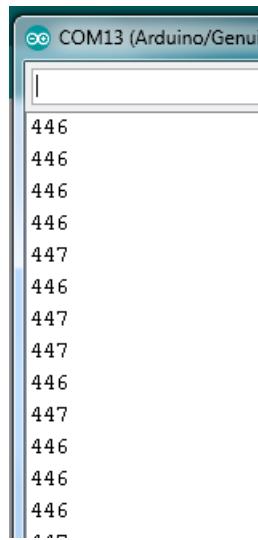
```

int sensorPin = A0;
void setup () {
    Serial.begin ( 9600 );
}
void loop () {
    int sensorValue = analogRead( sensorPin );
    Serial.println( sensorValue );
    delay ( 100 );
}

```

៤. លទ្ធផល

គីមិថុនា មិនបានបញ្ជូនតម្លៃ នៅពេលនេះ នៅពេលនេះ និងមានពន្លឹតម៉ែន ហើយអ្នកអាចយកតម្លៃទៅបញ្ជាកំពុលបាន ។



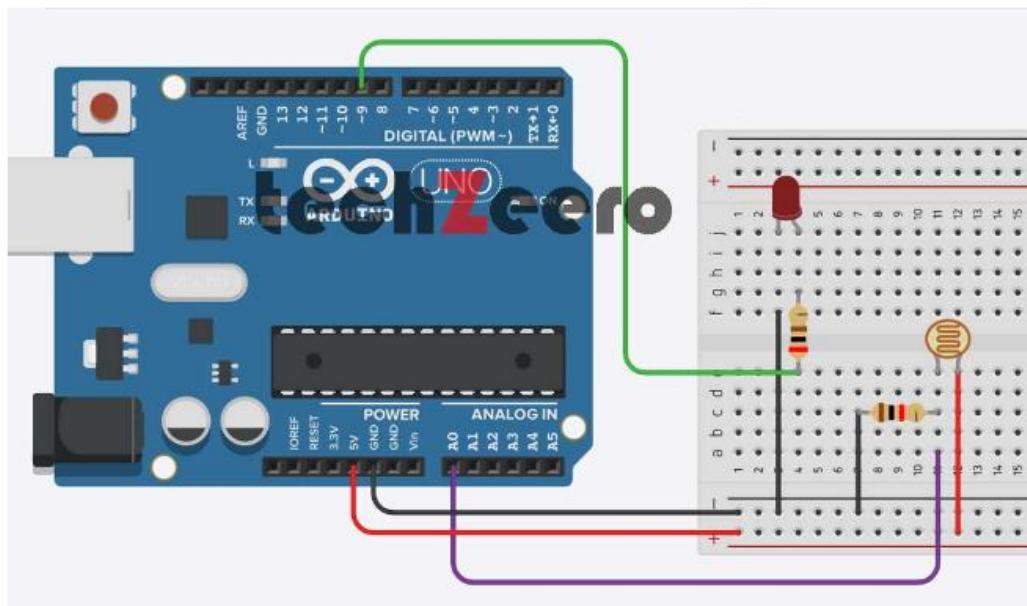
៥. តិោនាគន្ល់

យកតំលៃដែលចេញពី LDR ទៅបញ្ជាក់អំពុលទៅលើដឹងទី នៃ Arduino ក្រណីយកដែលបានបង្ហាញ។

❖ តម្រូវការគ្រែងសម្រាប់ពិសោធន៍

- Arduino Uno ១គ្រឹង
- USB Cable ១ខ្សែ
- LDR (light dependent resistor) ១គ្រាប់
- នសិស្សដែលមិនមែន 10 k Ω ១គ្រាប់
- នសិស្សដែលមិនមែន 330 Ω ១គ្រាប់
- LED ១គ្រាប់
- Breadboard ១ដ្ឋាន
- ខ្សែសំរាប់ត្រួតពិនិត្យ (Jumper wire) 7 ខ្សែ

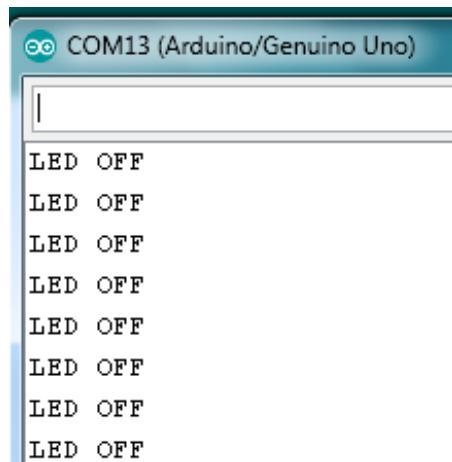
❖ ការត្រួតពិនិត្យសម្រាប់ពិសោធន៍



❖ ក្បាល់

```
int sensorPin = A0 ;  
int ledPin = 9 ;  
void setup () {  
Serial.begin(9600);  
pinMode(sensorPin , INPUT);  
pinMode(ledPin,OUTPUT);  
}  
void loop () {  
int sensorValue = analogRead(sensorPin);  
if(sensorValue <=300) {  
digitalWrite(ledPin,HIGH);  
Serial.print ("LED ON");  
Serial.println(sensorValue);  
delay (100);  
}  
else {  
digitalWrite(ledPin,LOW);  
Serial.print ("LED OFF");}  
}
```

❖ លទ្ធផល

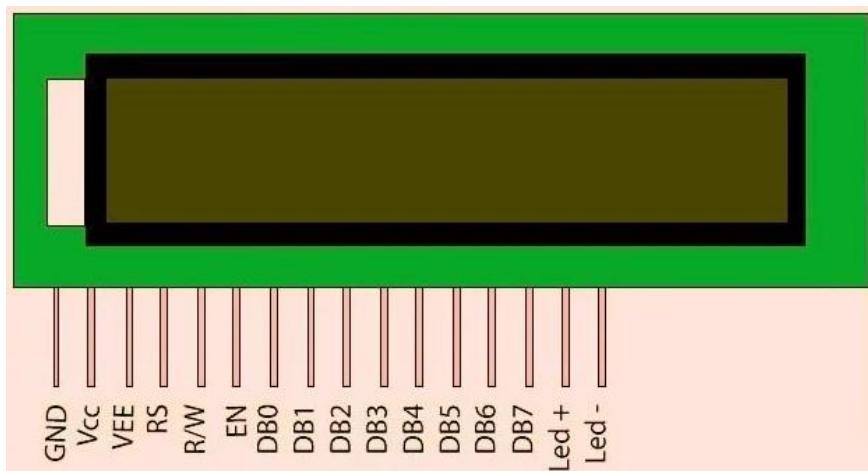


គិតថែទាំ ១១ : LCD display

LCD ជាប្រភេទខ្លួនដែលបានបង្កើតឡើងដោយក្រុមហ៊ុនក្រោមធម្មតានឹងបង្ហាញនិងបង្ហាញពាណិជ្ជកម្មនៃក្រុមហ៊ុនក្រោមធម្មតានឹងបង្ហាញ។



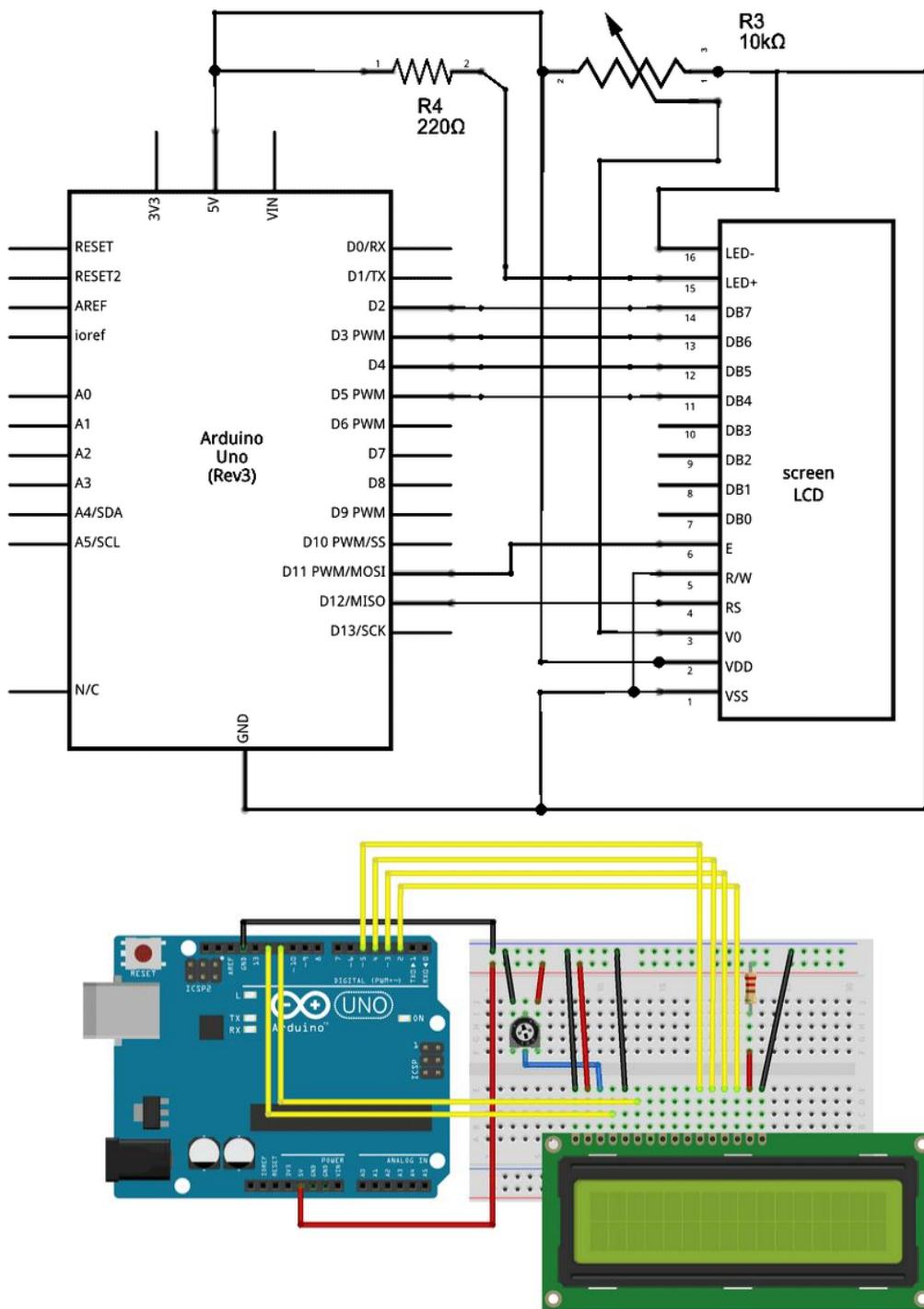
ក្នុងការពិសោធន៍កិច្ច នេះអ្នកនឹងសិក្សាអំពី LCD 16x2 ដែលមាននំយោកជាប្រភេទ LCD ដែលរាយបញ្ជាផ្ទាល់អ្នកបាន ២ដី និងមួយដីមេ ១៦ អក្សរអតិបរមា ។ ជាងម្ខាត LCD 16x2 មាន១៦ ដីដែលរាយនិមួយៗមានមុខងារដូចខាងក្រោម ៖



ដើម្បីអាចប្រើបានយើងត្រូវដាក់បញ្ជូន Library របស់ LCD ទៅក្នុង Sketch របស់យើងតាមរយៈ Sketch >> Include library >> LiquidCrystal ហើយយើងនឹងយើងត្រូវបង្រៀនក្នុង Sketch របស់យើង ៖ # include < LiquidCrystal.h >

ឧបនា ក្រោមគីឡូ និងក្នុងក្រុងម្រាប់កែស 16x2 LCD បង្ហាញថ្មីរោង Hello Svay Reang

១. សេចក្តីផ្តើម



២. ក្នុង

```

1 // include the library code:
2 #include <LiquidCrystal.h>
3
4 // initialize the library by associating any needed LCD interface pin
5 // with the arduino pin number it is connected to
6 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
7 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
8
9 void setup() {
10    // set up the LCD's number of columns and rows:
11    lcd.begin(16, 2);
12    // Print a message to the LCD.
13    lcd.setCursor(3, 0);           // move cursor to (3, 0)
14    lcd.print("Hello");          // print message at (3, 0)
15    lcd.setCursor(0, 1);           // move cursor to (0, 1)
16    lcd.print("Svay Rieng"); // print message at (0, 1)
17 }
18
19 void loop() {
20    // Turn off the display:
21    //lcd.noDisplay();
22    // delay(500);
23    // Turn on the display:
24    lcd.display();
25    delay(500);
26 }
27

```

៣. Function នៃជួលបច្ចេកទេសកូដការពេញសម្រាប់អ៊ីឡូឡូ

- ❖ LiquidCrystal ()
- ❖ begin ()
- ❖ clear ()
- ❖ home ()

- ❖ setCursor ()
- ❖ write ()
- ❖ print ()
- ❖ cursor ()
- ❖ noCursor ()
- ❖ blink ()
- ❖ noBlink ()
- ❖ display ()
- ❖ scrollDisplayLeft ()
- ❖ scrollDisplayRight ()
- ❖ autoscroll ()
- ❖ noAutoscroll ()
- ❖ leftToRight ()
- ❖ rightToLeft ()
- ❖ createChar()

➤ **LiquidCrystal ()** : ប្រើសម្រាប់ប្រកាសបឹកដើម Arduino អាយ LCD

Syntax :

LiquidCrystal (rs, enable, d4, d5, d6, d7)

LiquidCrystal (rs, rw, enable, d4, d5, d6, d7)

LiquidCrystal (rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal (rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

ឧត្តមានខ្លះ

```
#include<LiquidCrystal.h>
```

```
LiquidCrystal ( 12, 11, 10, 5, 4, 3,2);
```

```
void setup( ) {
```

```
    lcd.begin ( 16,1);
```

```
    lcd.print ("Hello");
```

```

    }
void loop () { }
```

- **begin ()** : ប្រើសម្រាប់ឱ្យ Library ចាប់ផ្តើមដំណឹកការ

Syntax :

```
lcd.begin ( cols, rows)
```

Parameters :

lcd : a variable of type liquidCrystal

cols : the number of columns that the display has

rows : number of rows that the display has

- **clear ()** : ប្រើសម្រាប់លើបអក្សរទាំងអស់នៅលើ screen

- **home ()** : ប្រើសម្រាប់ទាញ cursor ទៅកាន់ colum = 0 , Row = 0

- **setCursor ()** : ប្រើសម្រាប់ជាក់ Cursor ទៅកាន់ Colum , Row លាងមួយ

Syntax :

```
lcd.setCursor ( cols, rows)
```

- **write ()** : ប្រើសម្រាប់បញ្ជីនអក្សរមួយគ្នាដោយ LCD

Syntax :

```
lcd.write ( data)
```

ឧទាហរណ៍៖

```
#include<LiquidCrystal.h>
LiquidCrystal ( 12, 11, 10, 5, 4, 32);
void setup( ) {
    lcd.begin ( 16,1);
    lcd.write ("A");
}
void loop ( ) { }
```

- `print()` : ប្រើសម្រាប់បញ្ជាណក្សដែលមានតម្លៃនៅក្នុងទោកនៃ LCD

Syntax :

Lcd.print (data)

Lcd.print (data, BASE)

ឧទាហរណ៍៖

```
Lcd.print ("Hello");
```

Lcd.print (101,BIN);

Syntax :

lcd.createChar (num, data)

Parameters :

lcd : a variable of type liquidCrystal
num : which character to create (0 to 7)
rows : the character's pixel data

ឧទាហរណ៍៖

```
#include<LiquidCrystal.h>
LiquidCrystal ( 12, 11, 5, 4, 3,2);
Byte smiley [8] = { B00000,B10001,B00000,B00000,B10001,B01110,B00000,};

void setup( ) {
    lcd.createChar(0,smiley);
    lcd.begin (16,2);
    lcd.write(byte(0));
}

Void loop () {}
```

គិតថែទាំលេខ ១២ ៖ IR Remote



IR Remote វាគីជាកោតលេដលយើងប្រើរាល់ថ្វីជាមួយម៉ាសិនត្រជាក់ បុឡូទស្សន៍ ។ ក្នុងពិសោធន៍នេះយើងនឹងសិក្សាអំពីការចម្លងកំដូចជាការបង្កើតកោតលេនេះជាមួយArduino។ ដើម្បីអាចប្រើរាល់បានយើងត្រូវការ Library មួយដែលរកបានតាមរយៈ:

<https://github.com/Arduino-IRremote/Arduino-IRremote>

៩. Function នៃនេះ IR Remote ដែលបានបញ្ជាផ្ទាល់នៅក្នុងការបង្កើត

◆ ចំណោះដ្ឋីក ទឹក (Receiver)

- ❖ Constructor
- ❖ Decode Variable
- ❖ Begin
- ❖ decode
- ❖ get value
- ❖ resume()

➤ **Constructor :** នឹងប្រើប្រាស់ប្រកាសបើកដើរ IR remote

Syntax :

```
IRrecv ( IR pin) ;
```

ឧទាហរណ៍ ៖

```
IRrecv irrecv (8);
```

➤ **Decode variable**Syntax :

```
decode_results results;
```

ឧទាហរណ៍ ៖

```
IRrecv irrecv (8);
```

```
decode_results results;
```

➤ **begin**Syntax :

```
irrecv.enableIRIn ( ) ;
```

➤ **decode**Syntax :

```
irrecv.decode (&results);
```

Return :

True : if IR receive data from remote

False : if No data receive

➤ **Get Value**Syntax :

```
Results.value ;
```

Return :

Data from remote

➤ **resume ()**Syntax :

```
Irrecv.resume ( ) ;
```



នៅក្នុងពិសោធន៍ទី១បន្ថែមៗ យើងអត់យកមកពិសោធន៍ទេត្រឡាក់យើងប្រើតែល្អាប់ដើម្បី
យកទិន្នន័យដែលតែល្អាប់តែមេយោទៅលើផ្ទៀងផ្ទាត់ទូទៅ ហើយមាន Function របស់វាដែល
ជួយបញ្ជីបញ្ជាប់ក្នុងការសរសេរនិងពិសោធន៍របស់វា។ មានជាមឺន៖

- ❖ Constructor
- ❖ Send

Sketch For Receiver

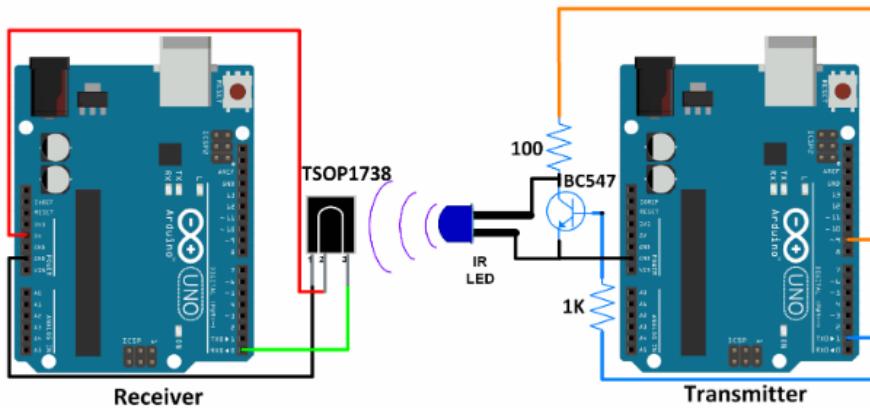
```
void setup() {
    Serial.begin(9600); /* Define baud rate for serial communication */
}

void loop() {
    if(Serial.available()) /* If data is available on serial port */
    {
        Serial.print(char(Serial.read())); /* Print character received on to the serial port */
    }
}
```

Receiver

Transmitter

IR Communication Between IR LED And IR Photodiode Using Arduino



IR Communication Between IR LED And TSOP1738 IR Receiver using Arduino

Sketch for Transmitter

```
void setup() {
    Serial.begin(9600); /* Define baud rate for serial communication */
}

void loop() {
    int count;
    for(count = 0; count<100; count++)
    {
        Serial.println(count);
        delay(1000);
    }
}
```

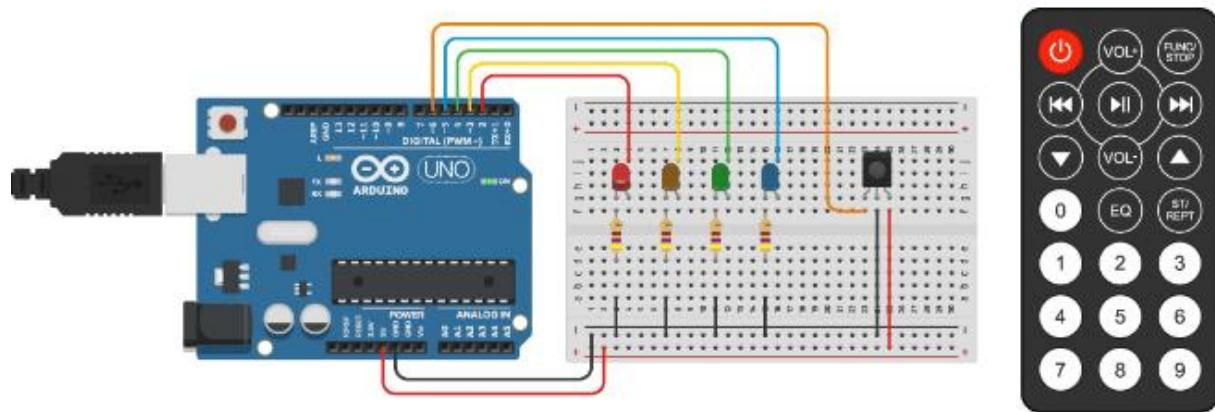


អនុវត្តន៍ សរស់រក្សាទបញ្ជាប់ LED តាមរយៈកែលរោយកីសិនរលក់បាន

ក្រឹមថាមពល

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. 330Ω resistor – 4 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED – 4 គ្រាប់
7. IR LED -1គ្រាប់
8. កែល -1គ្រឿង

សេចក្តីផ្តើម



កូដ

```
#include <IRremote.h>

#define RECEIVER_PIN 6 // define the connections
#define RED_LED_PIN 2
#define YELLOW_LED_PIN 3
#define GREEN_LED_PIN 4
```

```
#define BLUE_LED_PIN 5

IRrecv receiver(RECEIVER_PIN); // create a receiver object of the IRrecv class
decode_results results; // create a results object of the decode_results class
unsigned long key_value = 0;

void setup() {
    Serial.begin(9600); // begin serial communication at a baud rate of 9600
    receiver.enableIRIn(); // enable the receiver
    receiver.blink13(true); // enable blinking of the built-in LED when an IR signal is
    received

    pinMode(RED_LED_PIN, OUTPUT); // set the LED pins as output
    pinMode(YELLOW_LED_PIN, OUTPUT);
    pinMode(GREEN_LED_PIN, OUTPUT);
    pinMode(BLUE_LED_PIN, OUTPUT);

    digitalWrite(RED_LED_PIN, LOW); // turn all the LEDs off
    digitalWrite(YELLOW_LED_PIN, LOW);
    digitalWrite(GREEN_LED_PIN, LOW);
    digitalWrite(BLUE_LED_PIN, LOW);

}

void loop() {
    if (receiver.decode(&results)) {
        if (results.value == 0xFFFFFFFF) {
            results.value = key_value;
        }
        switch (results.value) {
            case 0xFD00FF:
                Serial.println("POWER"); break;
            case 0xFD807F:
                Serial.println("VOL+"); break;
        }
    }
}
```

```
case 0xFD40BF:  
    Serial.println("FUNC/STOP"); break;  
case 0xFD20DF:  
    Serial.println("I<<"); break;  
case 0xFDA05F:  
    Serial.println(">I"); break ;  
case 0xFD609F:  
    Serial.println(">>I"); break ;  
case 0xFD10EF:  
    Serial.println("DOWN"); break ;  
case 0xFD906F:  
    Serial.println("VOL-"); break ;  
case 0xFD50AF:  
    Serial.println("UP"); break ;  
case 0xFD30CF:  
    Serial.println("0"); break ;  
case 0xFDB04F:  
    Serial.println("EQ"); break ;  
case 0xFD708F:  
    Serial.println("ST/REPT"); break ;  
case 0xFD08F7:  
    Serial.println("1"); // run the ggle LED function with the red LED pin  
    toggleLED(RED_LED_PIN to); break ;  
case 0xFD8877:  
    Serial.println("2");  
    toggleLED(YELLOW_LED_PIN); break ;  
case 0xFD48B7:  
    Serial.println("3");
```

```
        toggleLED(GREEN_LED_PIN); break ;  
    case 0xFD28D7:  
        Serial.println("4");  
        toggleLED(BLUE_LED_PIN); break ;  
    case 0xFDA857:  
        Serial.println("5"); break ;  
    case 0xFD6897:  
        Serial.println("6"); break ;  
    case 0xFD18E7:  
        Serial.println("7"); break ;  
    case 0xFD9867:  
        Serial.println("8"); break ;  
    case 0xFD58A7:  
        Serial.println("9");  
        break ;  
    }  
    key_value = results.value;  
    receiver.resume ( ) ;  
}  
}  
  
void toggleLED(int pin) { // function to toggle the LED on and off  
    if (digitalRead(pin) == HIGH) { // if the LED is on  
        digitalWrite(pin, LOW); // turn it off  
    }  
    else { // else  
        digitalWrite(pin, HIGH); // turn it on  
    }  
}
```

ក្នុងបញ្ហា LED តាមរយៈគេលេខាយកី និងលត់បាន

(ក្រណីចុចលេខ 1, 2, 3 នៅលើគេលេ)

```
#include <IRremote.h>
#include <IRremoteInt.h>

#define code1 11111110011000011001111
#define code2 11111110001100011100111
#define code3 11111110111101010000101

int RECV_PIN = 3;
int led1 = 2;
int led2 = 4;
int led3 = 6;
int itsONled[ ] = {0,0,0,0};
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup(){
    Serial.begin(9600);
    irrecv.enableIRIn();
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
}

void loop(){
    if (irrecv.decode(&results)) {
        unsigned int value = results.value;
        switch(value) {
            case code1:
                if (itsONled[1] == 1) {
                    digitalWrite(led1,LOW);
                    itsONled[1] = 0;
                }
                if (itsONled[2] == 1) {
                    digitalWrite(led2,LOW);
                    itsONled[2] = 0;
                }
                if (itsONled[3] == 1) {
                    digitalWrite(led3,LOW);
                    itsONled[3] = 0;
                }
                break;
            case code2:
                if (itsONled[1] == 0) {
                    digitalWrite(led1,HIGH);
                    itsONled[1] = 1;
                }
                if (itsONled[2] == 0) {
                    digitalWrite(led2,HIGH);
                    itsONled[2] = 1;
                }
                if (itsONled[3] == 0) {
                    digitalWrite(led3,HIGH);
                    itsONled[3] = 1;
                }
                break;
            case code3:
                if (itsONled[1] == 0) {
                    digitalWrite(led1,HIGH);
                    itsONled[1] = 1;
                }
                if (itsONled[2] == 0) {
                    digitalWrite(led2,HIGH);
                    itsONled[2] = 1;
                }
                if (itsONled[3] == 0) {
                    digitalWrite(led3,HIGH);
                    itsONled[3] = 1;
                }
                break;
        }
    }
}
```

```
        }

        else{
            digitalWrite(led1,HIGH);
            itsONled[1] = 1; }

            break;

        case code2:
            if (itsONled[2] == 1){

                digitalWrite(led2,LOW);
                itsONled[2] = 0;

            }

            else{

                digitalWrite(led2,HIGH);
                itsONled[2] = 1; }

            break;

        case code3:
            if (itsONled[3] == 1){

                digitalWrite(led3,LOW);
                itsONled[3] = 0; }

            else{

                digitalWrite(led3,HIGH);
                itsONled[3] = 1;

            }

            break;

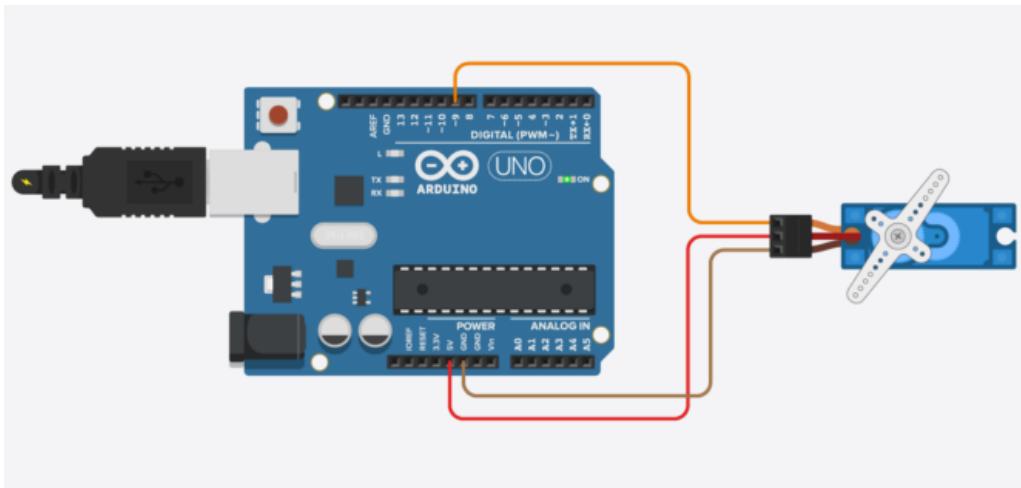
        }

        Serial.println(value);
        irrecv.resume();
    }
}
```

តិះសោរទំនើប ១៣ ៖ Servo Motor



អនុវត្តន៍ទី១



អាជីវិតរបាយកែងការណ៍នេះ:

<https://www.makerguides.com/servo-arduino-tutorial/>

ក្នុង:

```
// Include the servo library:  
#include <Servo.h>  
  
// Create a new servo object:  
Servo myservo;  
  
// Define the servo pin:
```

```
#define servoPin 9

// Create a variable to store the servo position:
int angle = 0;

void setup() {
    // Attach the Servo variable to a pin:
    myservo.attach(servoPin);

}

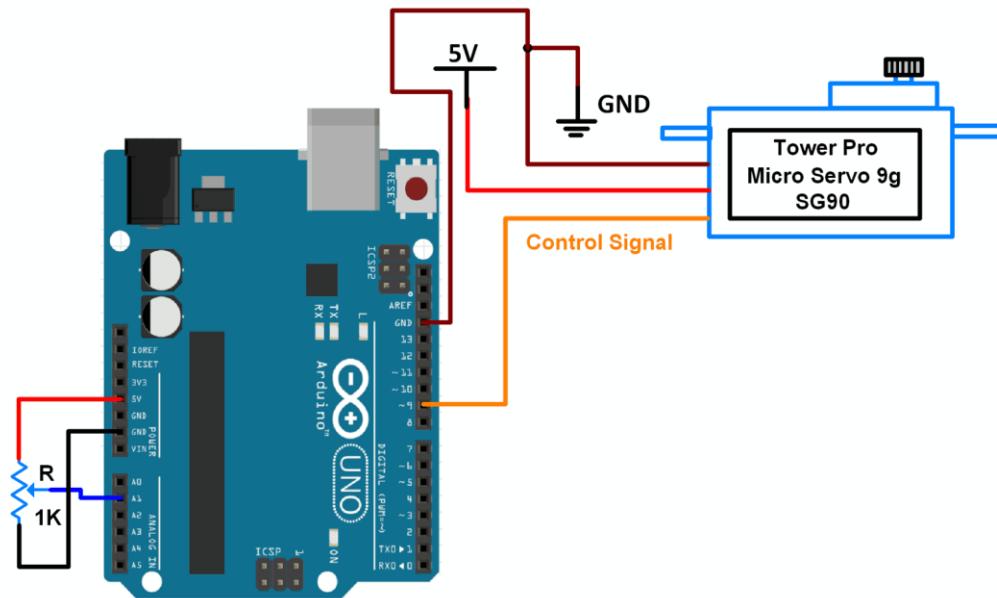
void loop() {
    // Tell the servo to go to a particular angle:
    myservo.write(90);
    delay(1000);
    myservo.write(180);
    delay(1000);
    myservo.write(0);
    delay(1000);

    // Sweep from 0 to 180 degrees:
    for (angle = 0; angle <= 180; angle += 1) {
        myservo.write(angle);
        delay(15);
    }

    // And back from 180 to 0 degrees:
    for (angle = 180; angle >= 0; angle -= 1) {
        myservo.write(angle);
        delay(30);
    }

    delay(1000);
}
```

អនុទត្តល់នឹង ប្រពើលាងកម្រិតថ្មី (វិវីឌី 1K)



អាចចូលតាមគេហទំនាក់ទំនងរបាយនេះ:

<https://www.electronicwings.com/arduino/servo-motor-interfacing-with-arduino-uno>

ប្រពើលាងប្រពើលិខិត

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. ធម្មិតស្ថានកូវិយ៉ាប់ 1K- 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. Servo motor
7. តែលប -1 គ្រឿង

កូដ

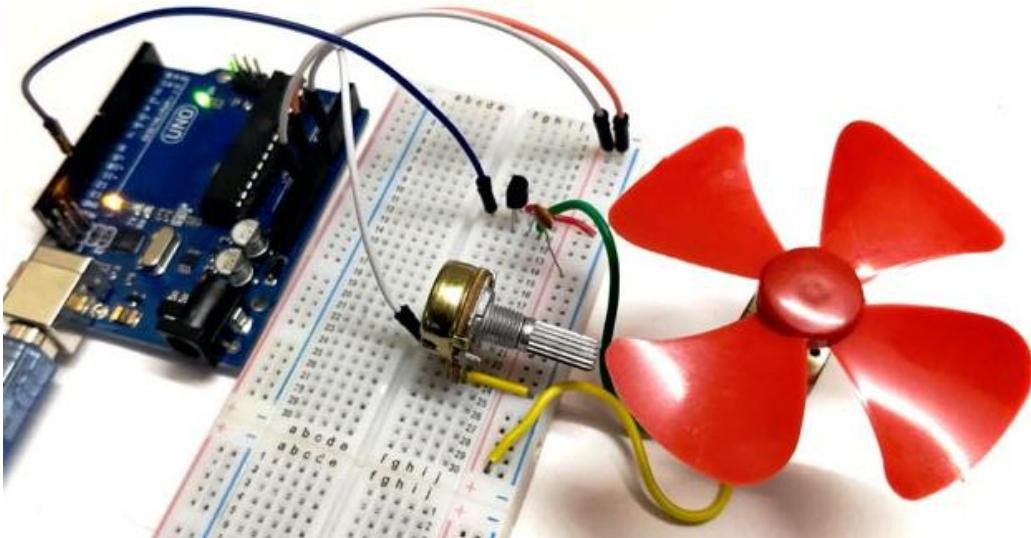
```
#include <Servo.h>

Servo myservo ; // create servo object to control a servo

int potpin = 1; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin
```

```
void setup() {  
    Serial.begin(9600);  
    myservo.attach(9); // attaches the servo on pin 9 to the servo object  
}  
  
void loop() {  
    val = analogRead(potpin); /* reads the value of the potentiometer  
    (value between 0 and 1023) */  
    Serial.print("Analog Value : ");  
    Serial.print(val);  
    Serial.print("\n");  
    val = map(val, 0, 1023, 0, 180); /* scale it to use it with the servo (value  
    between 0 and 180) */  
    Serial.print("Mapped Value : ");  
    Serial.print(val);  
    Serial.print("\n\n");  
    myservo.write(val); /* sets the servo position according to the  
    scaled value */  
    delay(1000); /* waits for the servo to get there */  
}
```

តិនោជល់ ១៤ ៖ DC Motor



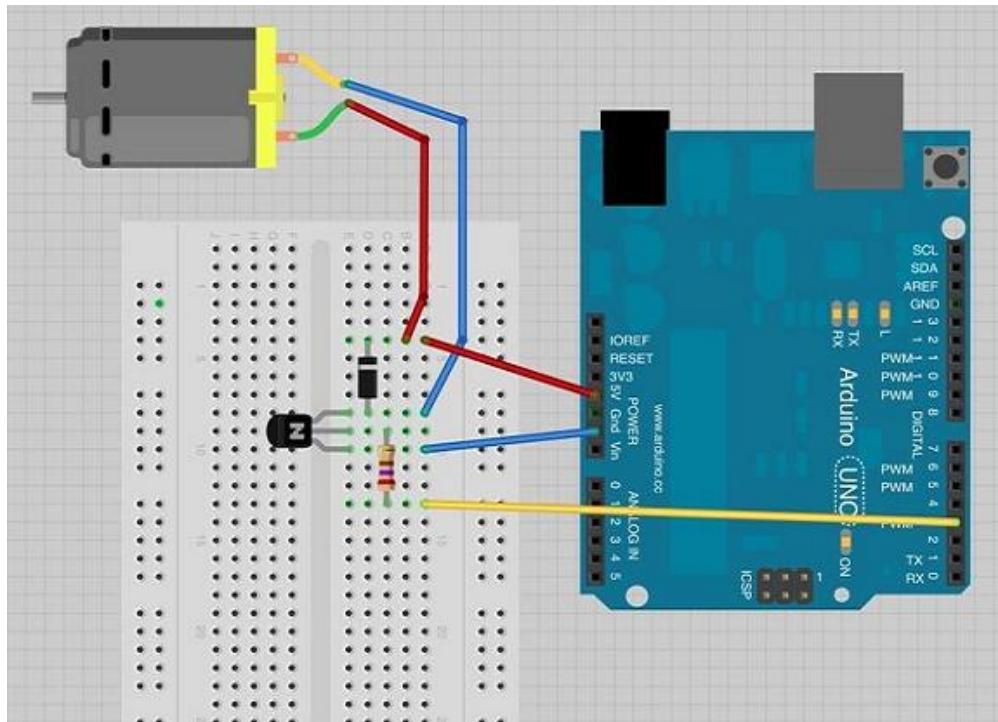
អាជីលតាមគេហទំន៽នខាងក្រោមនេះ:

https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm

ក្រុមសម្រួលនូវតម្លៃខាងក្រោម

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. ធមីស្ទឹងតម្លៃ 270Ω – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. DC motor – 1 គ្រាប់
7. ឱីយ៉ឺត ប្រភេទ 1N4001 -1 គ្រាប់
8. ត្រដឹងសុស្ស 2N2222A – 1 គ្រាប់

សេចក្តីផ្តើម



Spin Control Arduino Code

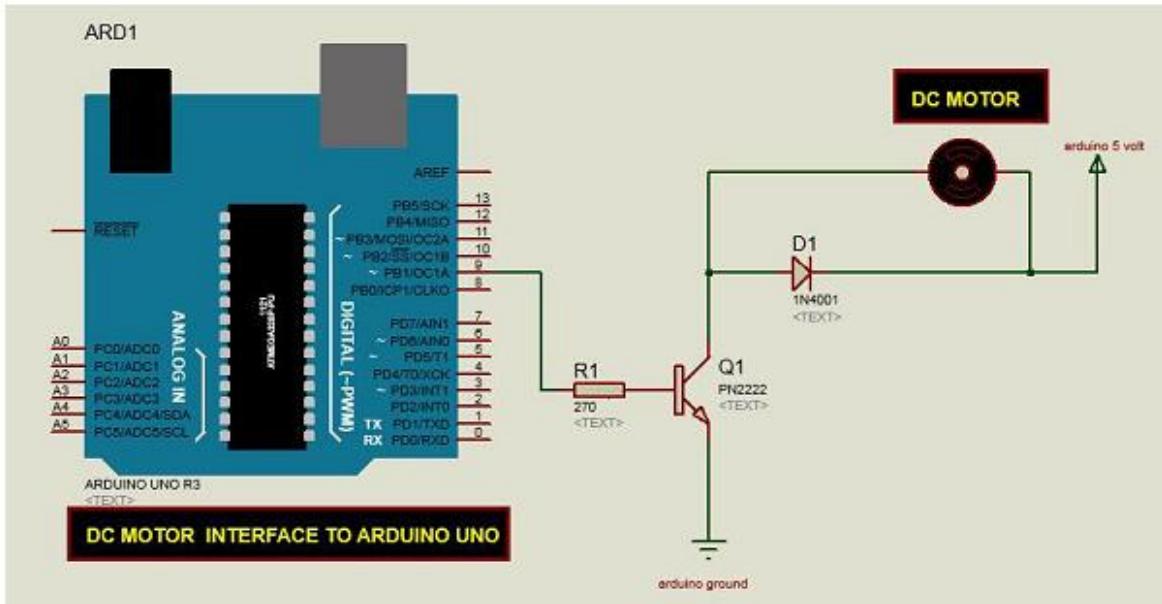
```
int motorPin = 3;

void setup() {
}

void loop() {
    digitalWrite(motorPin, HIGH);
}
```

Motor Speed Control

Following is the schematic diagram of a DC motor, connected to the Arduino board.



Arduino Code

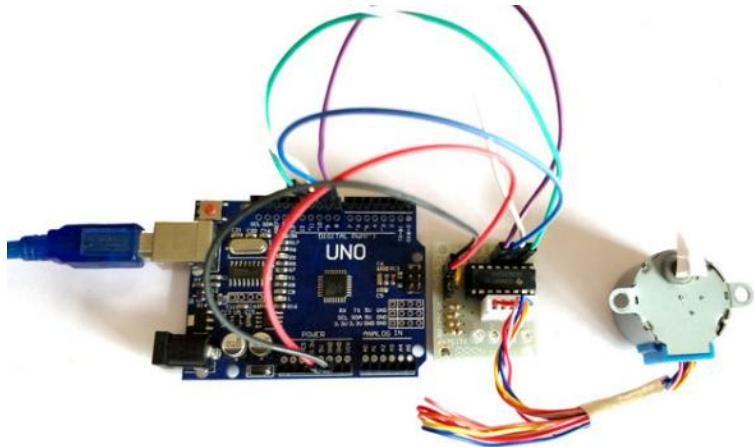
```
int motorPin = 9;

void setup() {
    pinMode(motorPin, OUTPUT);
    Serial.begin(9600);
    while (! Serial);
    Serial.println("Speed 0 to 255");
}

void loop() {
    if (Serial.available()) {
        int speed = Serial.parseInt();
        if (speed >= 0 && speed <= 255) {
            analogWrite(motorPin, speed);
        }
    }
}
```



តិនោបន់ទី ១ នៃ Step Motor



Connections :

- 9 (Arduino) ---> T1N1 (Motor driver)
- T1 (Arduino) ---> T1N2 (Motor driver)
- 10 (Arduino) ---> T1N3 (Motor driver)
- 6 (Arduino) ---> T1N4 (Motor driver)
- 12V+ ---> 5V (Arduino)
- 5V- ---> GND (Arduino)

កូន

```
#include <Stepper.h>

int numberofstep = 48*64;

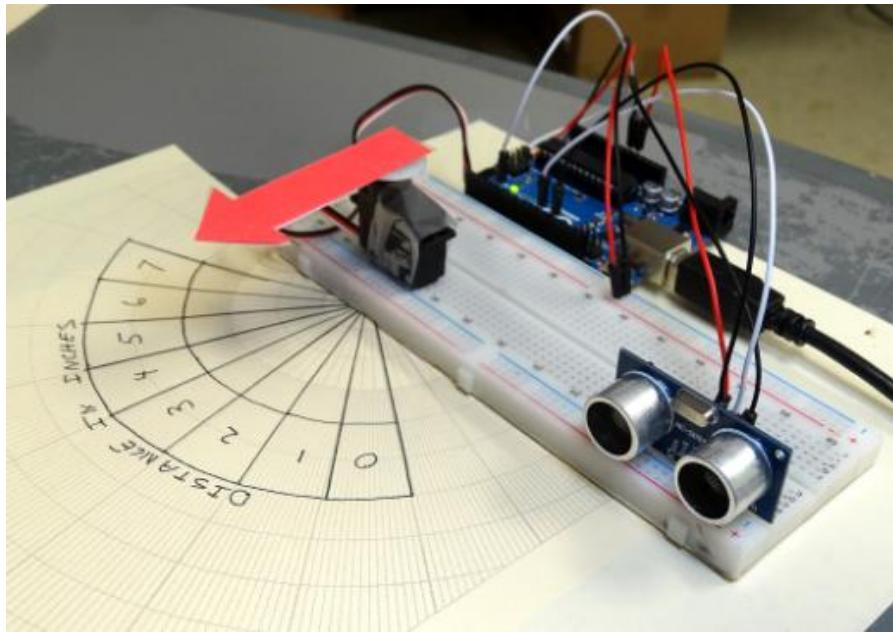
Stepper motor(numberofstep,9,11,10,6); // we use pins 9, 11, 10, 6

void setup() {
    motor.setSpeed(9); // we set motor speed at 9
}

void loop()
{
    motor.step(600); // the motor will run during 600 step (you can change the
    step)

    motor.step(-800); // the motor will run during 800 step in reverse (you can
    change the step)
}
```

តិះតេងលេខ ១៦ នៃ Distance Sensor



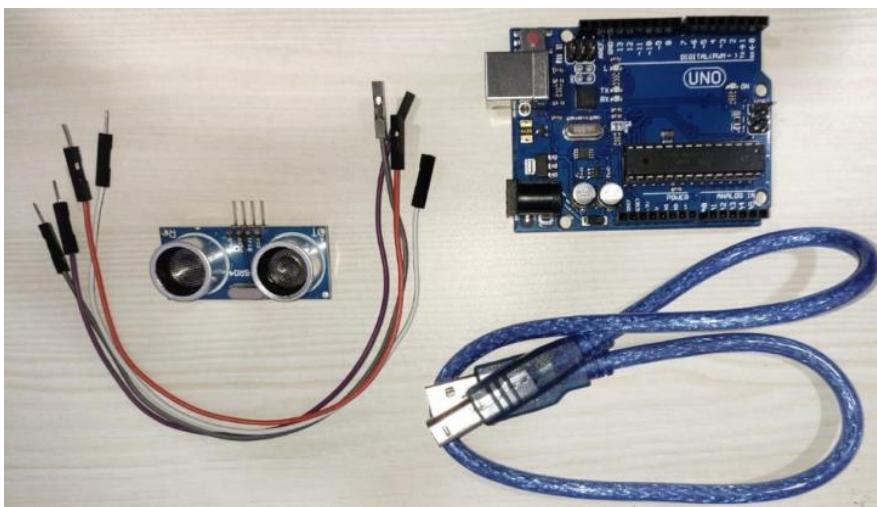
អនុវត្តន៍ទី១

រាយចក្របាលពាណិជ្ជកម្មនៃការងារនេះ:

<https://www.tutorialandexample.com/arduino-uno-distance-project/>

Requirements of hardware in this Project:

1. Arduino UNO board
2. USB cable connector for Arduino UNO
3. Ultra Sonic HC-SR04
4. Jumper wires male to female



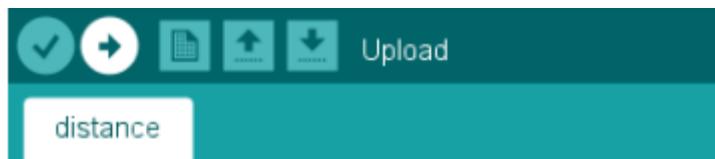
Ultrasonic Sensor HC-SR04 Arduino UNO

VCC -----> 5v

Trig -----> Pin 8

Echo -----> Pin 7

GND -----> GND



```
#include <Mouse.h>

const int trigpin= 8;
const int echopin= 7;
long duration;
int distance;
void setup(){
    pinMode(trigpin,OUTPUT);
    pinMode(echopin,INPUT);
    Serial.begin(9600);
}

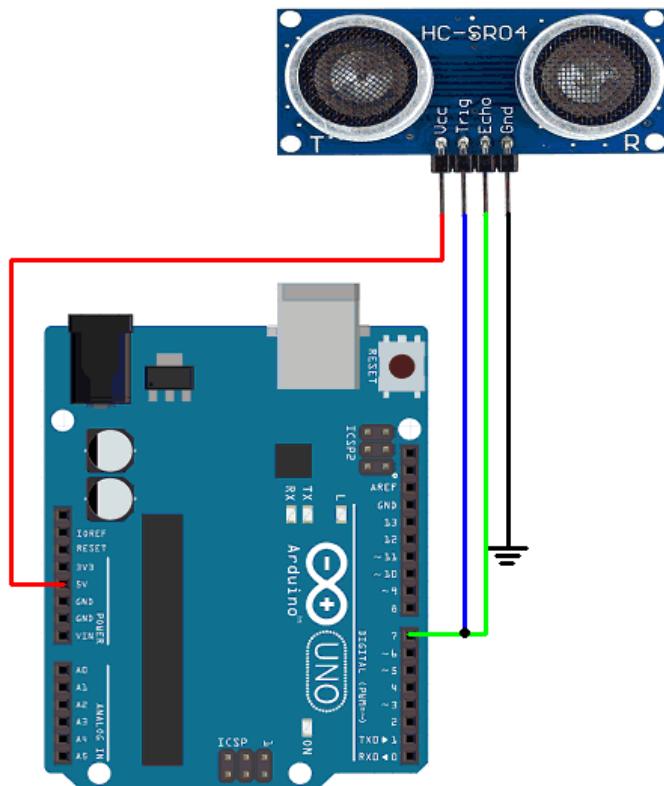
void loop(){
    digitalWrite(trigpin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigpin,LOW);
    duration=pulseIn(echopin,HIGH);
    distance = duration*0.034/2;
    Serial.println(distance);
}
```

តិនោជន៍ ១៦ : Distance Sensor

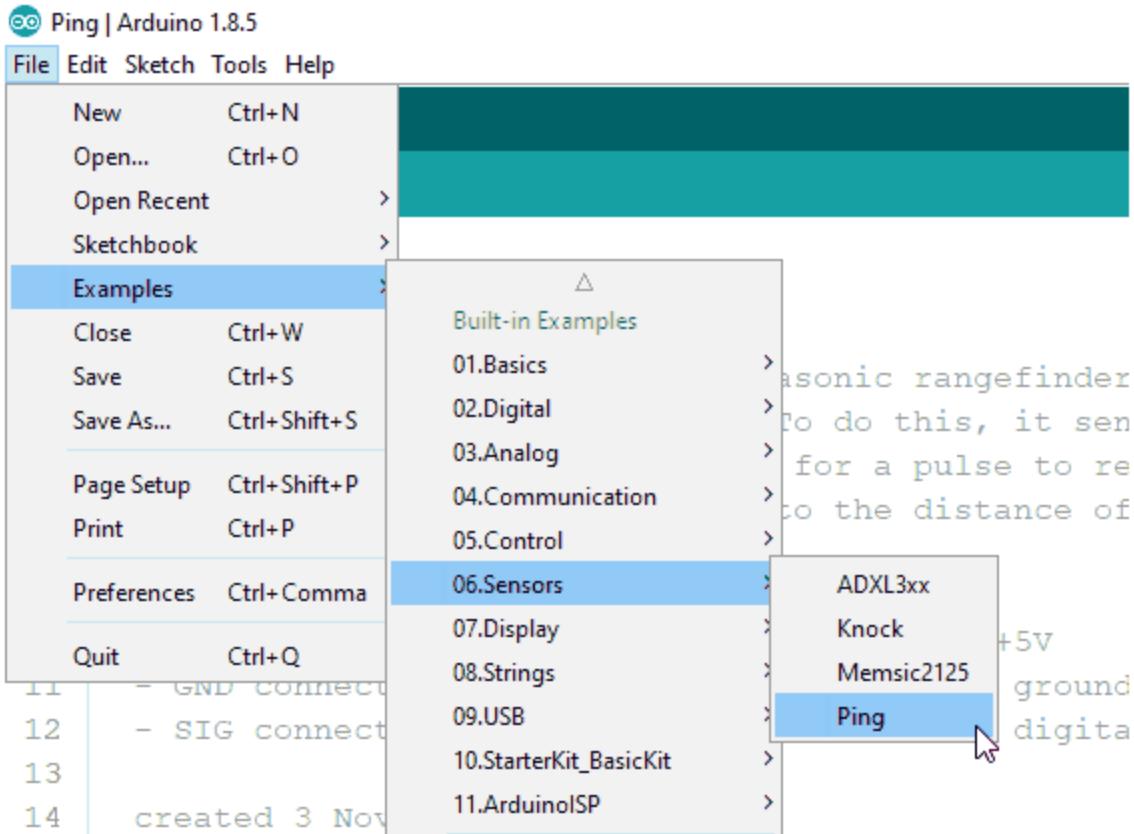
អនុទាុន័យឱ្យ



សេចក្តីផ្តើម



កូដ អាចចូលយកតាមប្រធានាផ្លាម



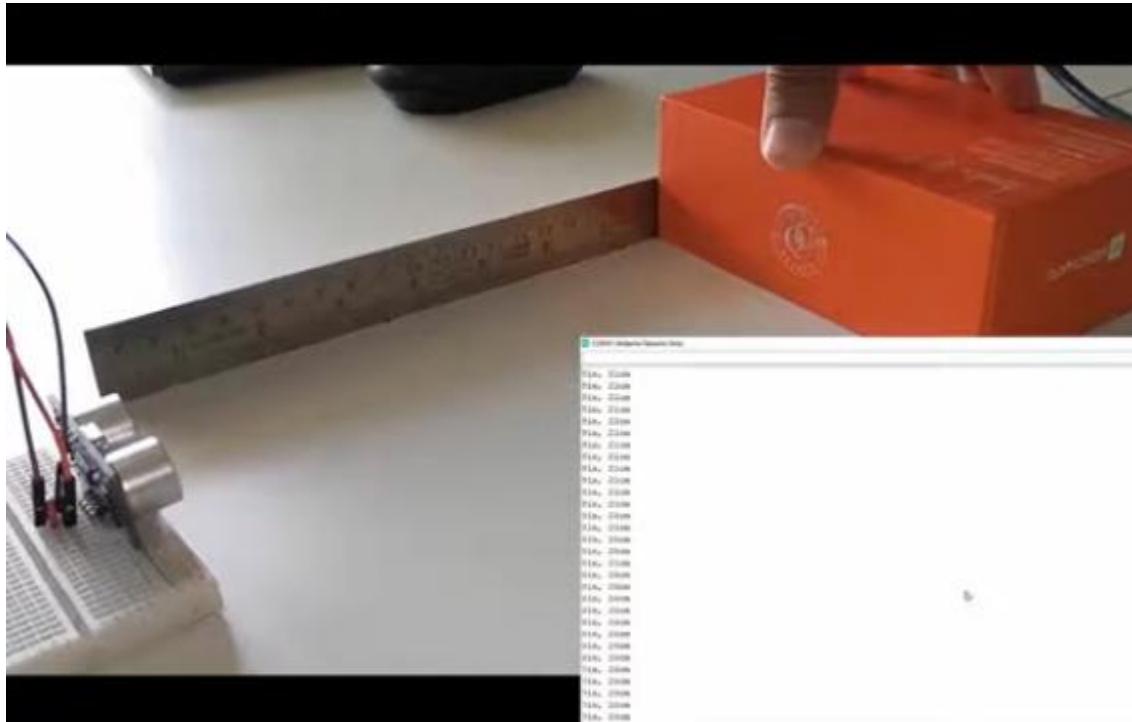
កូដ

```
// this constant won't change. It's the pin number of the sensor's output:  
const int pingPin = 7;  
  
void setup() {  
    // initialize serial communication:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // establish variables for duration of the ping, and the distance result  
    // in inches and centimeters:  
    long duration, inches, cm;
```

```
// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.  
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:  
  
pinMode(pingPin, OUTPUT);  
  
digitalWrite(pingPin, LOW);  
  
delayMicroseconds(2);  
  
digitalWrite(pingPin, HIGH);  
  
delayMicroseconds(5);  
  
digitalWrite(pingPin, LOW);  
  
// The same pin is used to read the signal from the PING))): a HIGH pulse  
// whose duration is the time (in microseconds) from the sending of the ping  
// to the reception of its echo off of an object.  
  
pinMode(pingPin, INPUT);  
  
duration = pulseIn(pingPin, HIGH);  
  
// convert the time into a distance  
  
inches = microsecondsToInches(duration);  
  
cm = microsecondsToCentimeters(duration);  
  
Serial.print(inches);  
  
Serial.print("in, ");  
  
Serial.print(cm);  
  
Serial.print("cm");  
  
Serial.println();  
  
delay(100);  
  
}  
  
long microsecondsToInches(long microseconds) {  
  
    // According to Parallax's datasheet for the PING))), there are 73.746  
    // microseconds per inch (i.e. sound travels at 1130 feet per second).  
    // This gives the distance travelled by the ping, outbound and return,  
    // so we divide by 2 to get the distance of the obstacle.
```

```
// See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf  
return microseconds / 74 / 2;  
}  
  
long microsecondsToCentimeters(long microseconds) {  
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.  
    // The ping travels out and back, so to find the distance of the object we  
    // take half of the distance travelled.  
    return microseconds / 29 / 2;  
}
```

លទ្ធផល

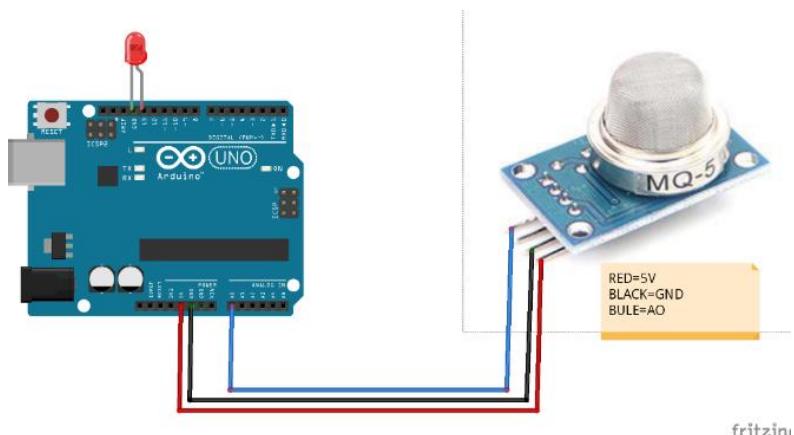


ទិន្នន័យលេខទី ១៧ : Gas Sensor

ក្រុមិតសង្គមអនុមតិតម្លៃទិន្នន័យ

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. Gas sensor (MQ-5) – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED

សេចក្តីផ្តើម



ក្នុង

```

const int gasPin = A0; //GAS sensor output pin to Arduino analog A0 pin

void setup() {
    Serial.begin(9600); //Initialize serial port - 9600 bps
}

void loop() {
    Serial.println(analogRead(gasPin));
    delay(1000); // Print value every 1 sec.
}

```

តិនោជន៍ ១៤ ៖ Keypad



៩. Function នៃ Keypad ដែលត្រូវបានបញ្ជាផ្ទាល់នៅក្នុងការសរស់សេរ

- ❖ Constructor
- ❖ waitForKey ()
- ❖ getKey ()
- ❖ getState ()
- ❖ setHoldTime ()
- ❖ setDebounceTime()
- ❖ addEventListener

➤ **Constructor :** ប្រើសម្រាប់ប្រកាសបើកដើម Library keypad

Syntax :

Keypad (makekeymap(userKeymap),row [], col [] , rows,cols)

ឧទាហរណ៍ ៦

```

const byte rows = 4; // four rows
const byte cols = 3; // three columns
char keys [rows] [cols] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'#', '0', '*'}
};

rowPins[rows] = {5,4,3,2}; //row pinout of the keypad
colPins[cols]={8,7,6}; //column pinout of the keypad

keypad = keypad(makeKeymap(keys),rowPins,colPins,rows,cols);

```

- **waitForKey** : ប្រើសម្រាប់រងចាថហូតទាល់ទៅចុចបូកធម្មុន ។ នៅពេលវាគើរកាល់ programme ទាំងអស់នឹងត្រូវ block រហូតដែល key ណាមួយត្រូវបានចុច
 - **getKey** : ប្រើសម្រាប់ទាញយកតំលៃលីតី key ណាមួយដែលយើងចុច
 - **getState** : ប្រើសម្រាប់មើលពីស្ថានភាពរបស់ key ថាគើរស្តិតនៅក្នុងសភាពធ្វើចម្លេច IDLE, PRESSED, RELEASED, HOLD
 - **setHoldTime ()** : ប្រើសម្រាប់កំណត់រយៈពេលគឺជាមីលវិនាទីដែលយើងត្រូវចុចបូកធម្មុនឲ្យ ជាប់រហូតដល់រយៈពេលនោះទើបអាចដំឡើការបាន

Syntax :

setHoldTime (unsigned int time)

- setDebounceTime () :
ក្រឹមសម្រាប់កំណត់រយៈពេលគិតជាមីនីនាទីដែលយើងត្រូវចួចបូកអានលក្ខណៈជាតុលាការ /noise

Syntax :

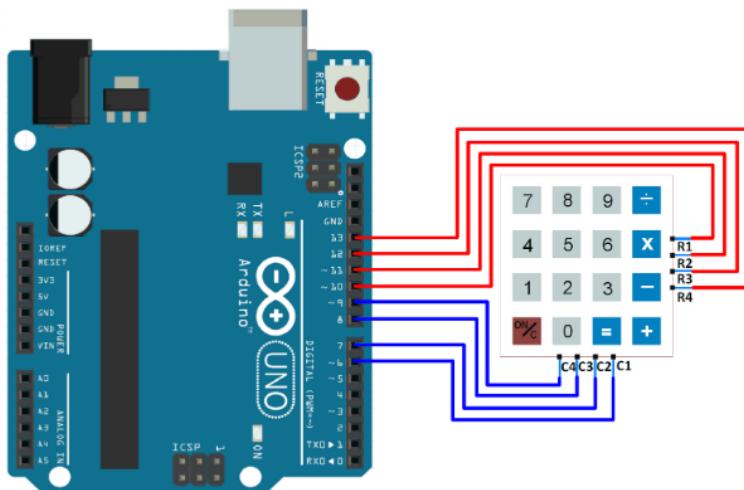
setDebounceTime (unsigned int time)

- `addEventListener ()` : ប្រើសម្រាប់នៅពេលដែល key ត្រូវបានចូច Function ចាប់ផ្តើមដំណើរការ

Syntax :

```
addEventListenner (keypad Event)
```

តែងតាំងនូវការ



Interfacing 4x4 Keypad with Arduino UNO

ក្រុម

```
#include <Keypad.h>

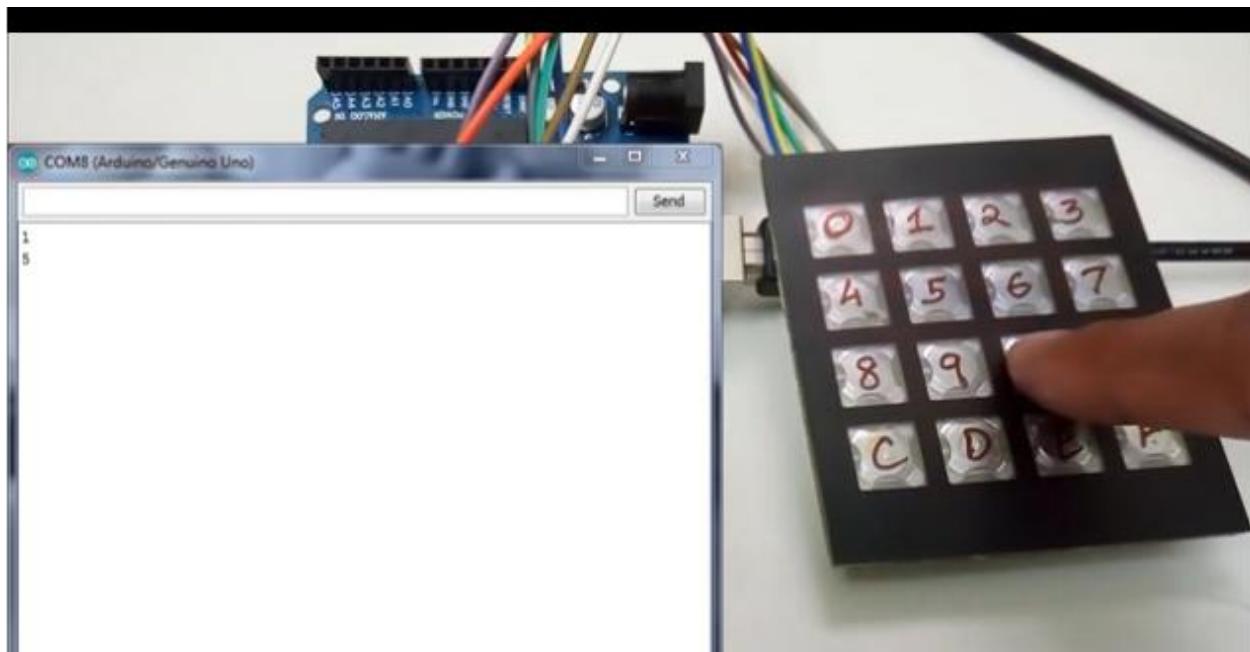
const byte ROWS = 4; /* four rows */
const byte COLS = 4; /* four columns */

/* define the symbols on the buttons of the keypads */
char hexaKeys[ROWS][COLS] = {

    {'0','1','2','3'},
    {'4','5','6','7'},
    {'8','9','A','B'},
    {'C','D','E','F'}
```

```
};  
  
byte rowPins[ROWS] = {10, 11, 12, 13}; /* connect to the row pinouts of the keypad */  
byte colPins[COLS] = {6, 7, 8, 9}; /* connect to the column pinouts of the keypad */  
/* initialize an instance of class NewKeypad */  
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS,  
COLS);  
void setup(){  
Serial.begin(9600);  
}  
void loop(){  
char customKey = customKeypad.getKey();  
if (customKey){  
Serial.println(customKey);  
}  
}
```

លទ្ធផល

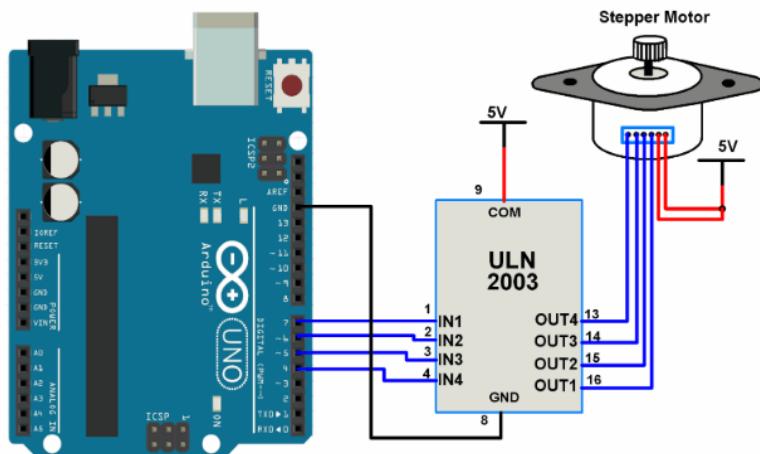


តិចោលទេសទំនើលេខ៖ Stepper Motor



មួយទៅ Stepper គឺជាមួយទៅ DC ដែលបែងចែកមុំបង្កិលពេញលេញនៃ 360° ទៅជាចំណុះនៃការបង្កិលបែងចែកនៃការរឹងរាល់បង្កិលដោយអនុវត្តលំដាប់ជាក់លាក់នៃសញ្ញាបញ្ញា។ លើពីនេះការបង្កិលអាចត្រូវបានធ្វានជាសំបុរាណដោយផ្តល់បញ្ហាអ្នកត្រូវដែលសញ្ញាបញ្ញា។

គេងអនុទំនើល



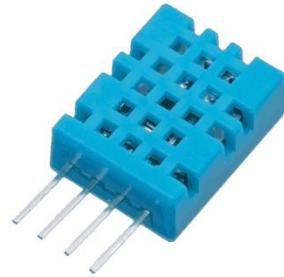
ក្នុង

```
void setup() {
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
}
```

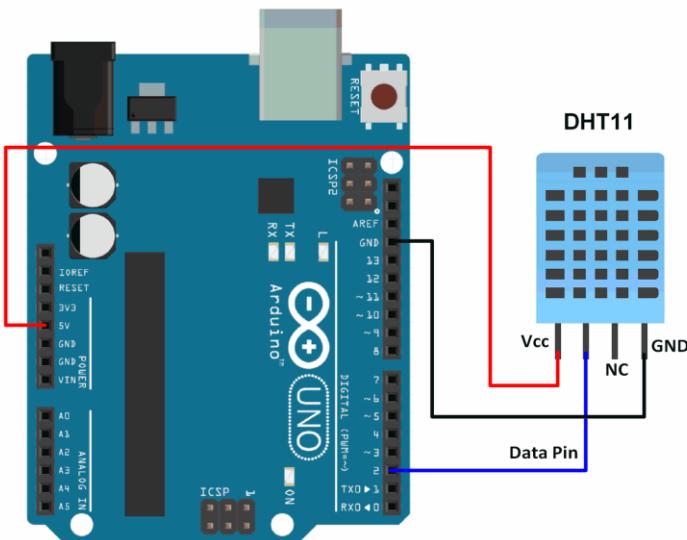
```
void loop() {  
    /* Rotation in one direction */  
    for(int i = 0; i<12; i++)  
    {  
        digitalWrite(7, HIGH);  
        digitalWrite(6, LOW);  
        digitalWrite(5, LOW);  
        digitalWrite(4, LOW);  
        delay(100);  
        digitalWrite(7, HIGH);  
        digitalWrite(6, HIGH);  
        digitalWrite(5, LOW);  
        digitalWrite(4, LOW);  
        delay(100);  
        digitalWrite(7, LOW);  
        digitalWrite(6, HIGH);  
        digitalWrite(5, LOW);  
        digitalWrite(4, LOW);  
        delay(100);  
        digitalWrite(7, LOW);  
        digitalWrite(6, HIGH);  
        digitalWrite(5, HIGH);  
        digitalWrite(4, LOW);  
        delay(100);  
        digitalWrite(7, LOW);  
        digitalWrite(6, LOW);  
        digitalWrite(5, HIGH);  
        digitalWrite(4, LOW);  
    }  
}
```

```
delay(100);  
digitalWrite(7, LOW);  
digitalWrite(6, LOW);  
digitalWrite(5, HIGH);  
digitalWrite(4, HIGH);  
delay(100);  
digitalWrite(7, LOW);  
digitalWrite(6, LOW);  
digitalWrite(5, LOW);  
digitalWrite(4, HIGH);  
delay(100);  
digitalWrite(7, HIGH);  
digitalWrite(6, LOW);  
digitalWrite(5, LOW);  
digitalWrite(4, HIGH);  
delay(100);  
}  
}
```

តិន្នន័យទី ២០ ៖ DHT11 Sensor



សេវាសម្បុទណ្ឌន៍



ដើម្បីរាជប្រឹកបាន យើងត្រូវការ Library មួយដែលរកបានតាមរយៈ:

<https://www.electronicwings.com/arduino/dht11-sensor-interfacing-with-arduino-uno>

ស្នើសុំ

```
#include "DHT.h"

DHT dht;

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

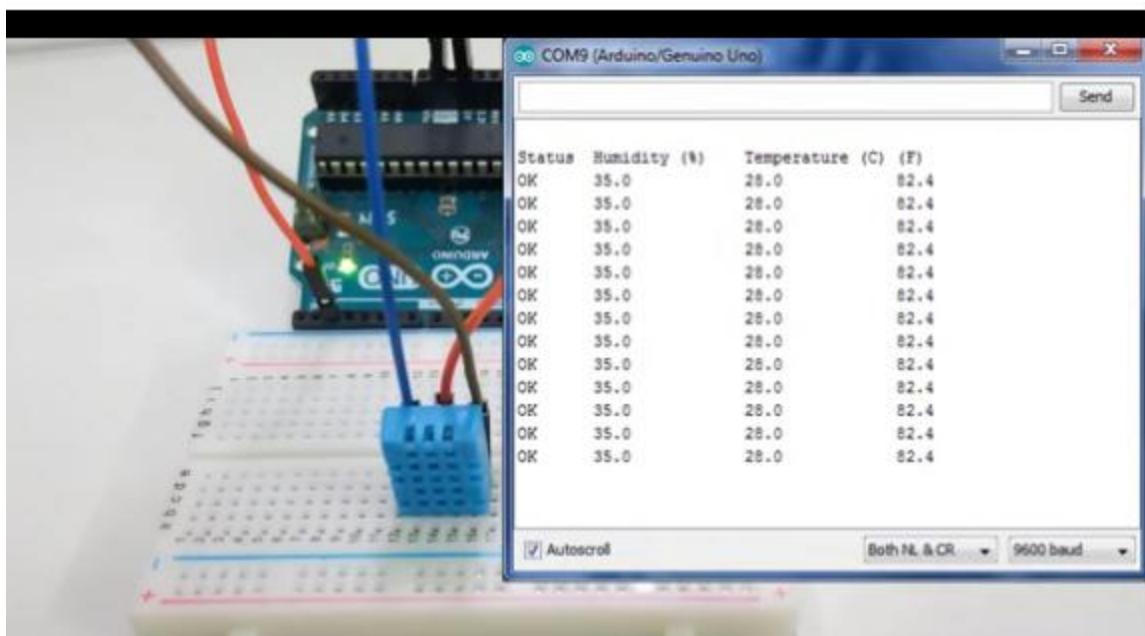
```

Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
dht.setup(2);                                /* set pin for data communication */
}

void loop() {
    delay(dht.getMinimumSamplingPeriod());      //Delay of amount equal to sampling
period
    float humidity = dht.getHumidity();          /* Get humidity value */
    float temperature = dht.getTemperature();     /* Get temperature value */
    Serial.print(dht.getStatusString());          /* Print status of communication */
    Serial.print("\t");
    Serial.print(humidity, 1);
    Serial.print("\t\t");
    Serial.print(temperature, 1);
    Serial.print("\t\t");
    Serial.println(dht.toFahrenheit(temperature), 1);    }

```

និត្យជន



គិតថែទាំ ២១ ៖ Alcohol (MQ-3) Sensor នៃទម្រង់ LCD និង buzzer

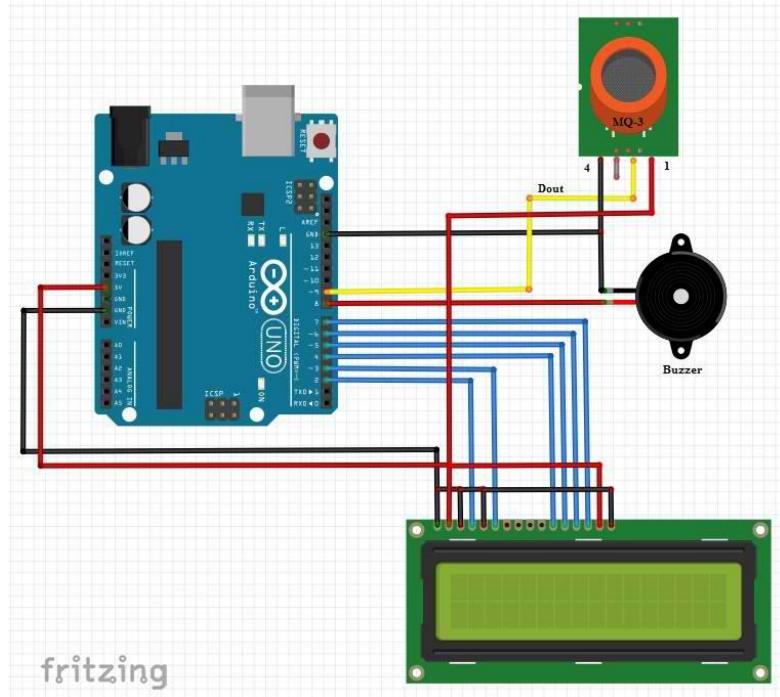
Alcohol sensor រាជ្យប្រភេទ Senso នឹងម្រាប់រាសកំវិតជាតិភាកុល ។ រាជ្យ OUTPUT ពីរប្រភេទ គឺ Digital និង Analog ដែលមាននៅមួយចំណេះអាចប្រើ DigitalRead និង AnalogRead បានឡើតាមដើរបស់រាជ្យបានកំណត់ ។



ក្រុមឈាមប្រព័ន្ធលើនៅខាងក្រោម

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. Alcohol sensor (MQ-3) – 1 គ្រាប់
4. Jumper wires
5. Breadboard
6. LED
7. LCD (16x2)
8. Buzzer

សេចក្តីផ្តើម



ក្នុង

```
#include <LiquidCrystal.h>

#define sensor 9 // Connnect MQ-3 Sensor to pin-9
#define buzzer 8 // Connnect Buzzer to pin-8

LiquidCrystal lcd(2, 3,4,5, 6,7);

// LCD PIN - ARDUINO PIN
// RS      -  2
// RW      -  GND
// E       -  3
// D4     -  4
// D5     -  5
// D6     -  6
// D7     -  7

void setup() {
```

```
lcd.begin(16, 2);
pinMode(sensor, INPUT);
pinMode(buzzer, OUTPUT);
lcd.clear();
lcd.print(" Arduino Alcohol");
lcd.setCursor(0,1);
lcd.print(" Sensor    ");
delay(2000);
}

void sens(){
int i=1;
lcd.clear();
lcd.print("Alcohol Detected");
digitalWrite(buzzer, HIGH);
}

void scn(){
lcd.clear();
lcd.print(" Sensing.... ");
digitalWrite(buzzer, LOW);
}

void loop()
{
if(digitalRead(sensor)==0)
sens();
else
scn();
}
```

លិខោដល់ ៤២ ៖ Interfacing HC-05 Bluetooth Module

HC-05 Bluetooth Module



HC-05 Bluetooth Module

- Operating Voltage : 4 V to 6V (have internal 3.3V regulator).
- Operating Current : 30mA
- Integrated antenna and an edge connector.
- Range about 10 meters.
- Configurable in both master and slave modes.
- Pins : STATE, RXD, TXD, GND, VCC, KEY/ENABLE

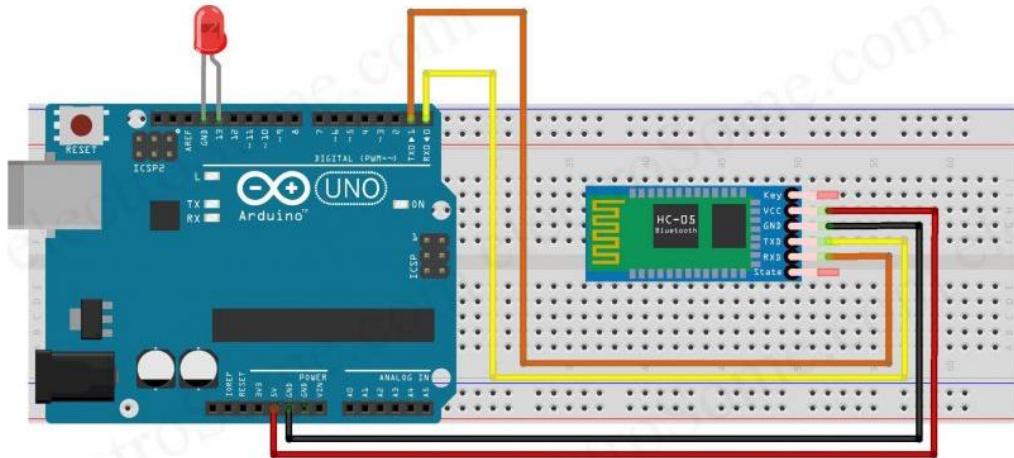
ផ្តល់នយោបាយ

1. Arduino Uno R3 with ATmega328P
2. USB cable
3. HC-05 Bluetooth Module – 1 គ្រប់
4. Jumper wires
5. Breadboard
6. LED

Software នយោបាយ

1. Arduino IDE
2. Android App (បង្កើត្យបច្ចេកទេសកូមិកភាពកោដ្ឋាក់សនស្តាយរៀង)
ឈ្មោះរបស់វា គឺ Arduino bluetooth controller អាចចូលតាម link ខាងក្រោម៖
<https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor>

សេចក្តីផ្តើម



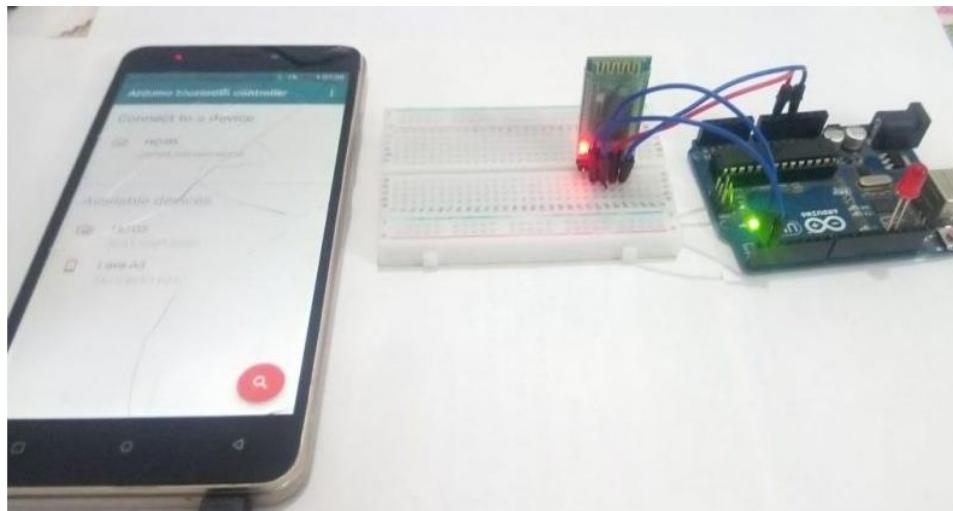
គឺជា

```

char data = 0;           //Variable for storing received data
void setup() {
    Serial.begin(9600); //Sets the data rate in bits per second (baud) for serial data
    transmission
    pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
}
void loop() {
    if(Serial.available() > 0) { // Send data only when you receive data:
        data = Serial.read(); //Read the incoming data and store it into variable data
        Serial.print(data); //Print Value inside data in Serial monitor
        Serial.print("\n"); //New line
        if(data == '1') //Checks whether value of data is equal to 1
            digitalWrite(13, HIGH); //If value is 1 then LED turns ON
        else if(data == '0') //Checks whether value of data is equal to 0
            digitalWrite(13, LOW); //If value is 0 then LED turns OFF
    }
}

```

លទ្ធផល



ជំហានតែនលទ្ធផលទៅលើការការពារ

❖ PIR Motion Sensor

វាតាប្រភេទ Sensor ដែលធ្វើការនៅពេលដែលមានវត្ថុ បុរាណ: មានដឹរិតធ្លាស់ទីមានចលនា ពីមុខវា ។ ជាជម្យតារាយអោយច្រកចេញ (OUTPUT) ជាទីដឹងលើ (1 or 0) ។ ចំពោះកូដរបស់វាតីប្រើ digitalRead ដើម្បីទាប់យកសិរីញ្ញាល់ ។



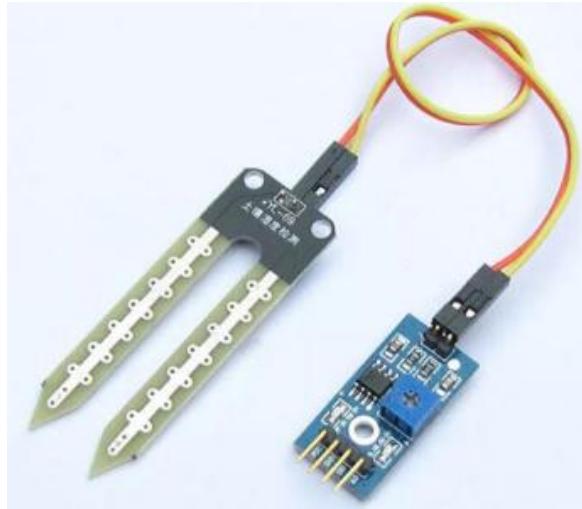
❖ Flame Sensor

វាតាប្រភេទ Sensor ប្រើសម្រាប់វាស់ប្រើសម្រាប់ចាប់យកសិរីញ្ញាល់ពេលមានភ្លើនអត្ថិភាព កើតឡើង ។ វាទាប់យកតែពន្លឹមណាតភ្លើន ហើយមិនចាប់យកផ្សេងៗនៅទេ។ វាមាន OUTPUT ពីរប្រភេទ គឺ Digital និង Analog ដែលមាននឹងយចាយឱ្យអាចប្រើ DigitalRead និង AnalogRead បានទៅតាមដើរបស់វាបានកំណត់ ។



❖ Soil Sensor

វាគារប្រគល់ Sensor ដើម្បីតាមតម្លៃស្ថាយដែលត្រូវបានគេប្រើសម្រាប់វាសំរាប់មិនសំឡោងដើម្បីបាន។ ទេ។ វាមាន OUTPUT ពីរប្រគល់ គឺ Digital និង Analog ដើម្បីមាននំយចាយឱ្យអាចប្រើ DigitalRead និង AnalogRead បានឡើតាមដើរបស់វាបានកំណត់។ ហើយតម្លៃប្រស់វាលើស្តី 1023 នៅពេលគ្មានសំឡោងនិងលើស្តី 0 នៅពេលដឹសីម ១០០% ។



❖ Rain Sensor

ប្រើសម្រាប់បានយកសីត្វាល់នៅពេលមានតំណាក់ទីកត្រួចប្រកាសកំចាំរី។ វាអាយ OUTPUT ជាតម្លៃ Digital ។



ចូលនឹងលំនៅតិចខាងក្រោម

លោកស្រីនស្រីនស្រីន

១. ចូរគូសស្រីតិចប់តាម Current sourcing (ការយកចរន្តចញពី UNO)

និងបង្ហាញលក្ខណៈដែលការបស់វា?

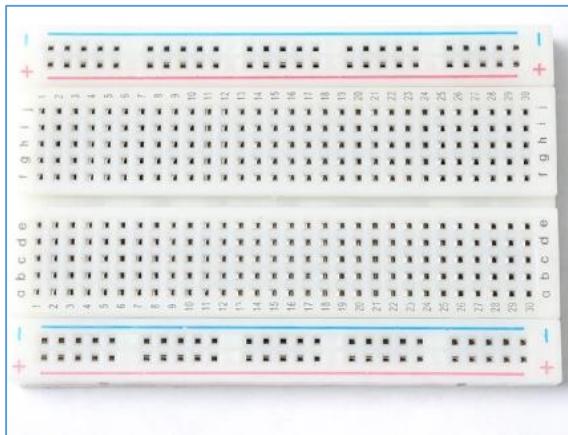
២. ចូរគូសស្រីតិចប់តាម Current sinking (ការយកចរន្តចូល UNO)

និងបង្ហាញលក្ខណៈដែលការបស់វា?

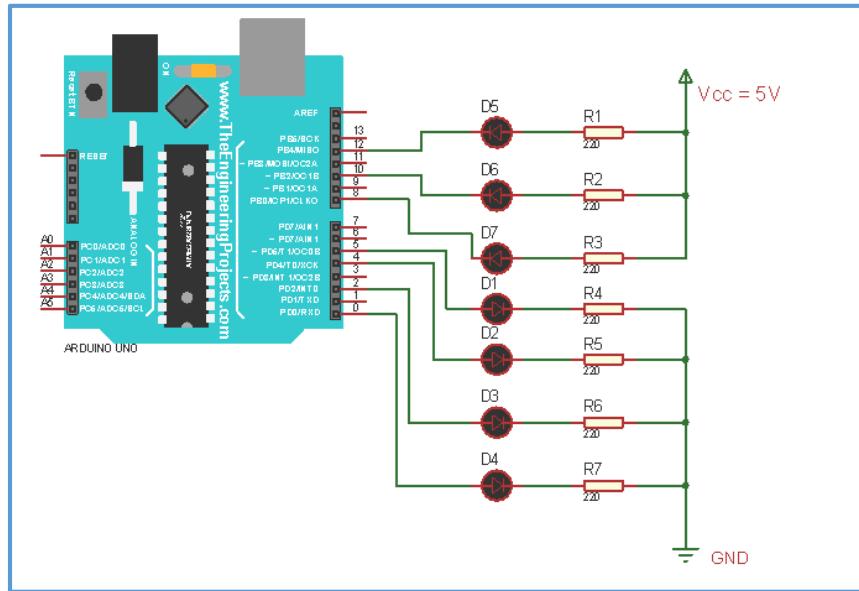
៣. ចូរគូសស្រីតិចប់តាម Pull-up Resistor និងបង្ហាញលក្ខណៈដែលការបស់វា?

៤. ចូរគូសស្រីតិចប់តាម Pull-down Resistor និងបង្ហាញលក្ខណៈដែលការបស់វា?

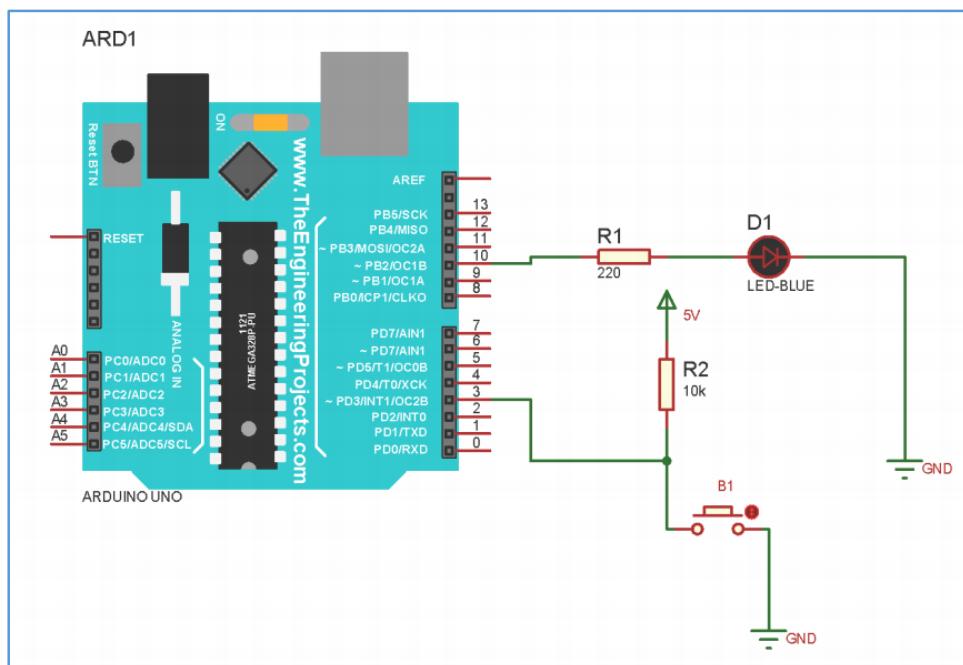
៥. ចូរពន្លឺដែលបានក្រោម (Bread Board) ដូចម្របខាងក្រោម ?



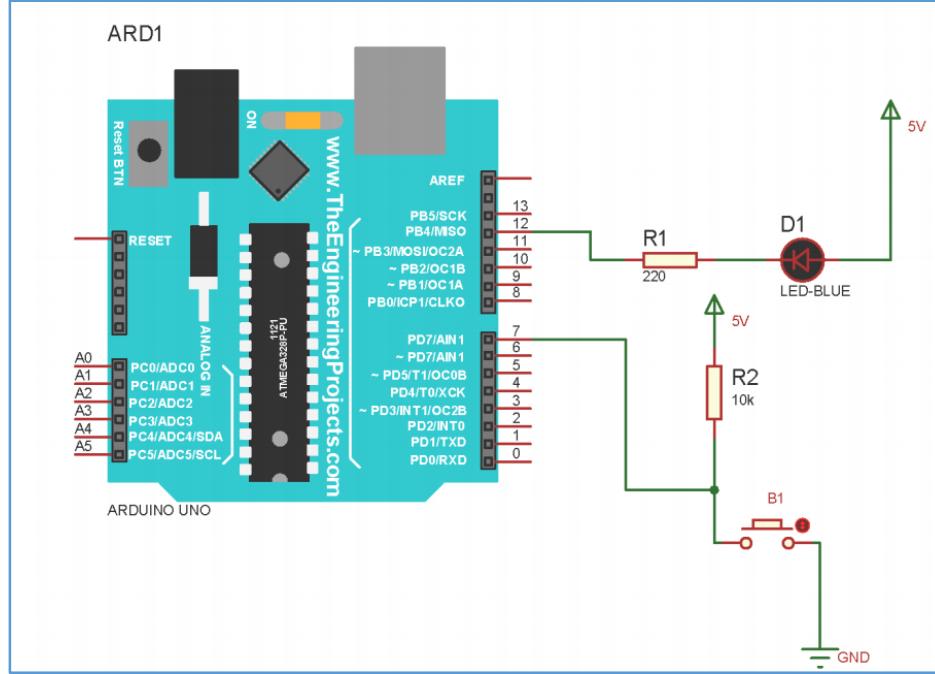
៦. ចូរសរស់រកដួចខាងក្រោម លក្ខណៈដំណឹកការគី អោយ LED តីទាំងអស់ ?



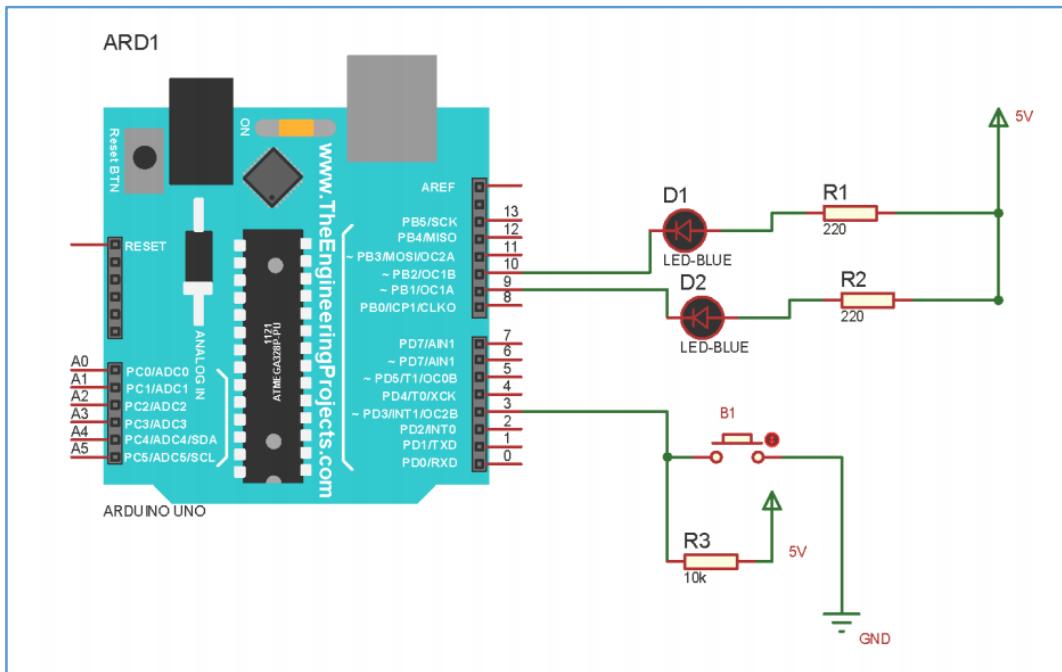
៧. ចូរសរស់រកដួចខាងក្រោម លក្ខណៈដំណឹក ករណីចុច B1 នាំចូ D1 តី ករណីអតិចុច B1 នាំចូ D1 អតិថិជ្ជ



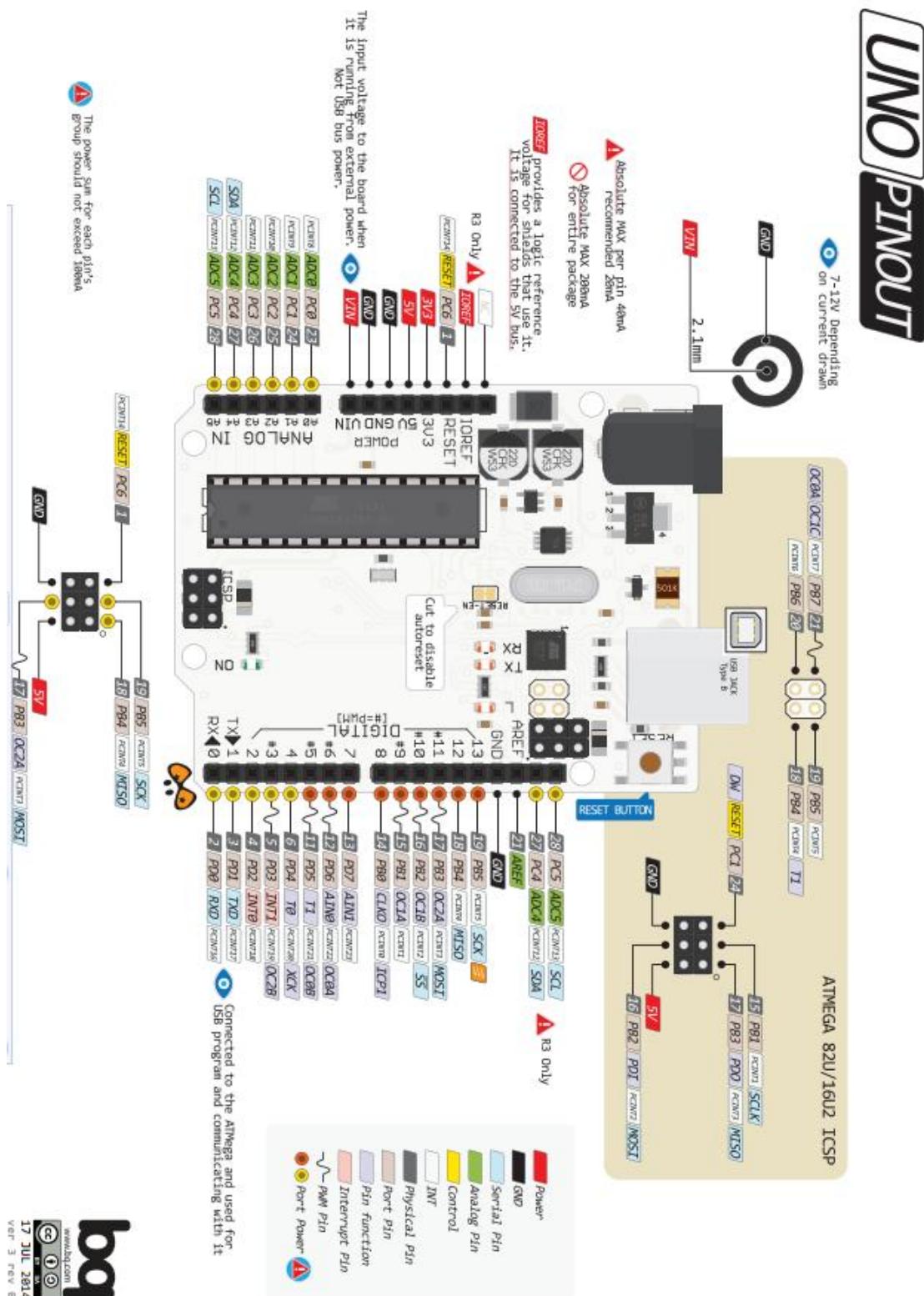
៤. ចូរសរស់រក្សាទិន្នន័យក្នុងខាងក្រោម លក្ខណៈដំណើរ ករណីចុច B1 នាំចូរ D1 នឹង ករណីអតិថិជ្ជ B1 នាំចូរ D1 អតិថិជ្ជ



៥. ចូរសរស់រក្សាទិន្នន័យក្នុងខាងក្រោម លក្ខណៈដំណើរ ករណីចុច B1 នាំចូរ D1 និង D2 អតិថិជ្ជ ករណីអតិថិជ្ជ B1 នាំចូរ D1 និង D2 អតិថិជ្ជ



Datasheet



ឯកសារយោង

ឯកសារចំណាំសៀវភៅ មួយចំនួនដូចខាងក្រោម៖

១. Hacking Electronics- Learning Electronics with Arduino and Raspberry Pi, 2nd Edition

២. Make: learn electronics with Arduino (JODY CULKIN and ERIC HAGAN)

ឯកសារចំណាំរបាយការយោងមួយចំនួនដូចខាងក្រោម៖

<https://www.arduino.cc/en/Main/Software>

<https://github.com/Arduino-IRremote/Arduino-IRremote>

<https://www.makerguides.com/servo-arduino-tutorial/>

<https://www.electronicwings.com/arduino/servo-motor-interfacing-with-arduino-uno>

https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm

<https://www.tutorialandexample.com/arduino-uno-distance-project/>

<https://www.electronicwings.com/arduino/dht11-sensor-interfacing-with-arduino-uno>

<https://electrosome.com/interfacing-hc-05-bluetooth-module-arduino-uno/>