

4

ພື້ນຖານຂອງ Arduino

4.1 ແນະນຳກ່ຽວກັບ Arduino

Arduino ໄດ້ຖືກປະດິດຢູ່ທີ່ Ivrea Interaction Design Institute. ມັນໄດ້ຖືກອອກແບບສໍາລັບການ prototyping ໄວ, ເປົ້າໝາຍ hobbyist ໂດຍມີພື້ນຖານການຂຽນໂປຼແກຼມໃດໆ. ທັນທີທີ່ແພລະຕະຟອມທີ່ເປັນມິດກັບຜູ້ໃຊ້ໄດ້ດຶງດູດຜູ້ຊົມທີ່ກວມເອົາຊຸມຊົນທີ່ກວ້າງຂວາງແລະເລີ່ມປ່ຽນແປງເພື່ອເປັນຕົວແນວໂນ້ມຫລ້າສຸດໃນຕະຫຼາດ, ຈາກກະດານ 8-bit ກັບຜະລິດຕະພັນ IoT, ອຸປະກອນທີ່ໃສ່ໄດ້, ແລະສະພາບແວດລ້ອມຝັງຕົວ. ກະດານ Arduino ແມ່ນແຫຼ່ງເປີດຢ່າງສົມບູນແລະສາມາດໃຊ້ສໍາລັບການພັດທະນາແອັບພລິເຄຊັນທີ່ມີຄວາມຕ້ອງການໂດຍສະເພາະ. ຊອບແວ Arduino ແມ່ນເປັນມິດກັບຜູ້ໃຊ້ແລະງ່າຍທີ່ຈະເລີ່ມຕົ້ນດ້ວຍສະພາບແວດລ້ອມທີ່ມີຄວາມຍືດຫຍຸ່ນສໍາລັບຜູ້ໃຊ້ຂັ້ນສູງ. ມັນສາມາດດໍາເນີນການໄດ້ໃນເວທີ Mac, Linux, ແລະ Window. ສິ່ງໃຫມ່ສາມາດຮຽນຮູ້ກັບ Arduino.

ຂໍ້ ດີ ຂອງ Arduino :

ຄ່າໃຊ້ຈ່າຍ:ກະດານ Arduino ແມ່ນລາຄາແພງກວ່າເມື່ອທຽບກັບຈຸນລະພາກອື່ນໆ.
ກະດານຄວບຄຸມ.

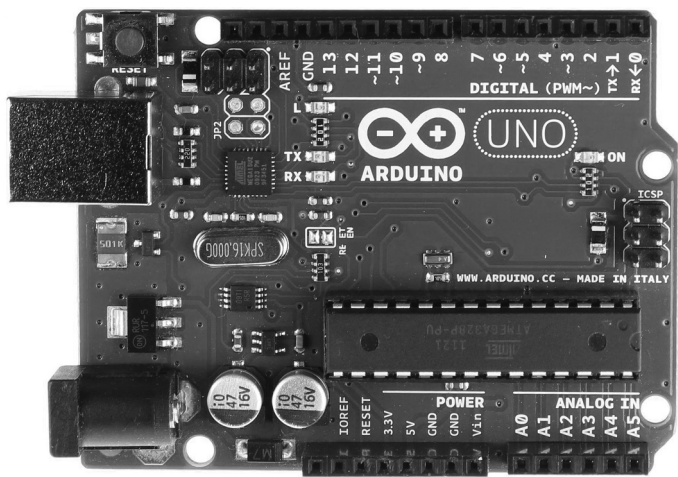
ເວທີ:ຊອບແວ Arduino (IDE) ແມ່ນເຫມາະສົມກັບສ່ວນໃຫຍ່ຂອງ
ລະບົບປະຕິບັດການເຊັ່ນ Macintosh OSX, Windows, ແລະ Linux.

ເປັນມິດກັບຜູ້ໃຊ້:ຊອບແວ Arduino (IDE) ແມ່ນເປັນມິດກັບຜູ້ໃຊ້, ງ່າຍທີ່ຈະ
ເລີ່ມຕົ້ນ, ແລະມີຄວາມຍືດຫຍຸ່ນສໍາລັບນັກຂຽນໂປລແກລມທີ່ມີຄວາມຊໍານິຊໍານານ.

ແຫຼ່ງເປີດ:Arduino ເປັນຊອຟແວແຫຼ່ງເປີດທີ່ສາມາດເປັນ
ໂປຣແກມດ້ວຍພາສາ C, C++, ຫຼື AVR-C. ດັ່ງນັ້ນຄວາມຫລາກຫລາຍຂອງໂມດູນ
ສາມາດອອກແບບໂດຍຜູ້ໃຊ້.

4.1.1 Arduino Uno

Arduino/Genuino Uno ມີ microcontroller ATmega328 onboard. ມັນມີ 6 ຜອດ ປ້ອນຂີ້ມູນແບບອະນາລັອກ (A0–A5) ແລະ 14 ຜອດ I/O ດິຈິຕອລ, ໃນນັ້ນ 6 ຮູບສຽງ PWM. ແຕ່ລະ pin ສາມາດດໍາເນີນການກ່ຽວກັບ 0-5 V ຂອງແຮງດັນ. ມັນດໍາເນີນການຢູ່ທີ່ 16 MHz ຂອງຄວາມຖີ່. [ຮູບທີ 4.1](#) ສະແດງກະດານ Arduino Uno ([ຕາຕະລາງ 4.1](#)).



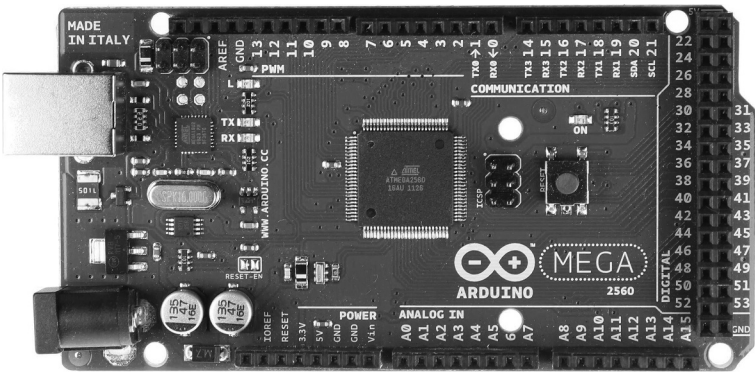
ຮູບທີ 4.1
ກະດານ Arduino Uno.

ຕາຕະລາງ 4.1
ລາຍລະອຽດ PIN ຂອງ Arduino UNO

ປັນຍາດ	ລາຍລະອຽດ
ວິນ	ມັນເປັນແຮງດັນພາຍນອກໃຫ້ກັບກະດານ 3.3 V
3.3 ວ	ການສະຫນອງ, ໃນຄະນະ
+ 5 ວ	ແຮງດັນຂາອອກ +5 V
GND	Ground
IOREF	ມັນແມ່ນການເລືອກແຫຼ່ງພະລັງງານທີ່ເຫມາະສົມໂດຍການສະຫນອງແຮງດັນອ້າງອີງ
ລຳດັບການຂັດຂວາງພາຍນອກ	ມັນ ສາ ມາດ ສົງ ແລະ ຮັບ ຂໍ້ ມູນ serial ດ້ວຍ 0(Rx) 1(Tx) ສົງ ຜົນ ໃຫ້ເກີດ ການ ຂັດ ຂວາງ ໃນ ຄ່າ 0 ັ້ າ (pins 2 ແລະ 3)
PWM	8 bit ທົກ PWM (3, 5, 6, 9, 10, 11)
SPI	ມັນສະຫນັບສະຫນູນການສື່ສານ SPI [10 (SS), 11 (MOSI), 12 (MISO) ແລະ 13 (SCK)]
LED	inbuilt LED ຂັບເຄື່ອນ
TWI	ການສື່ສານ TWI [A4 (SDA), ແລະ A5 (SCL)] ແຮງດັນໄຟຟ້າ
AREF	ອ້າງອີງກັບວັດສະດຸປ້ອນການປຽບທຽບ
ຮີເຊັດ	ມັນຖືກນຳໃຊ້ເພື່ອປັບ microcontroller onboard

4.1.2 Arduino Mega

Arduino Mega ມີ microcontroller ATmega2560 onboard. ມັນມີ 16 ອະນາລັອກອິນພຸດ, 54 ດິຈິຕອລ I/Os, ການເຊື່ອມຕໍ່ USB, 4 UART, ຊ່ອງສຽບໄຟ, ແລະປຸ່ມຮີເຊັດ. ມັນດຳເນີນການກ່ຽວກັບຄວາມຖີ່ 16 MHz.ຮູບທີ 4.2ສະແດງກະດານ Arduino Mega (ຕາຕະລາງ 4.2).



ຮູບທີ 4.2
ກະດານ Arduino Mega.

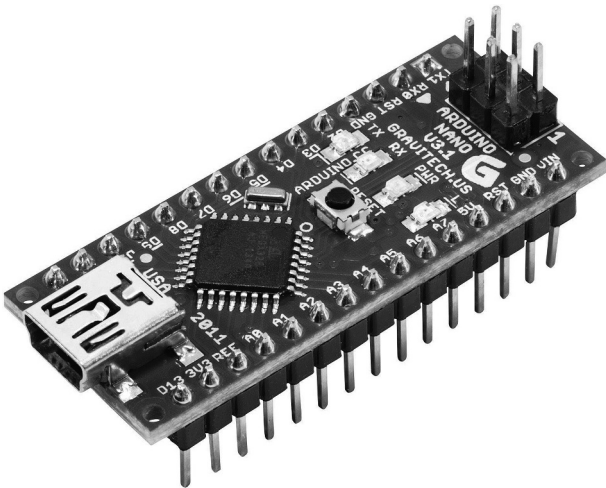
ຕາຕະລາງ 4.2

ບົກໝູດຄຳອະທິບາຍ

ລາຍລະອຽດ	
ບົກໝູດ	
ວິນ	ແຮງດັນພາຍນອກຂອງກະດານ Arduino ອອກເປັນ 5 V
+ 5 ວ	ຄວບຄຸມ
3.3 ວ	Onboard 3.3 V supply
GND	Ground
IOREF	ມັນແມ່ນການເລືອກແຫຼ່ງພະລັງງານທີ່ເໝາະສົມໂດຍການສະຫນອງແຮງດັນອ້າງອີງ
ລຳດັບ 0	ມັນສາມາດສົ່ງແລະຮັບຂໍ້ມູນ serial ດ້ວຍ 0(Rx) ແລະ 1(Tx) ມັນສາມາດສົ່ງແລະຮັບ
ລຳດັບ 1	ຂໍ້ມູນ serial ດ້ວຍ 19(Rx) ແລະ 18(Tx) ມັນສາມາດສົ່ງແລະຮັບຂໍ້ມູນ serial
ລຳດັບ2	ດ້ວຍ 14(Rx) ແລະ 16(Tx))
ການຂັດຂວາງພາຍນອກ	ມັນກະຕຸ້ນການລົບກວນພາຍນອກຢູ່ທີ່ຄ່າຕົ້ນດ້ວຍ 2 (interrupt 0), 3 (ຂັດຂວາງ 1), 18 (ຂັດຂວາງ 5), 19 (ຂັດຂວາງ 4), ແລະ 20 (ຂັດຂວາງ 2)
PWM	PWM 8 ບິດ (pins: 2–13 ແລະ 44–46)
SPI	ມັນສະຫນັບສະຫນູນການສື່ສານ SPI [10 (SS), 11 (MOSI), 12 (MISO), ແລະ 13 (SCK)]
LED	LED ຂັບເຄື່ອນຢູ່ທີ່ pin 13
TWI	ຮອງຮັບການສື່ສານ TWI [pins: 20 (SDA), 21 (SCL)] ມັນເປັນແຮງດັນ
AREF	ໄຟຟ້າອ້າງອີງສໍາລັບການປ້ອນຂໍ້ມູນການປຽບທຽບ
ຮີເຊັດ	ມັນຖືກນໍາໃຊ້ເພື່ອປັບ microcontroller ໃນກະດານ

4.1.3 Arduino Nano

Arduino/Genuino Nano ມີ microcontroller ATmega328 onboard. ມັນມີ 8 ອະນາລັອກອິນພຸດ, 14 ພອດ I/O ດິຈິຕອນ, ແລະ 6 PWM. ມັນມີຫນ່ວຍຄວາມຈໍາ flash 32 KB, 1 KB EEPROM, 2 KB SRAM, ແລະເຮັດວຽກຢູ່ທີ່ 16 MHz ຂອງຄວາມຖີ່.ຮູບທີ 4.3ສະແດງໃຫ້ເຫັນ Arduino Nano (ຕາຕະລາງ 4.3ແລະ4.4).



ຮູບທີ 4.3
ກະດານ Arduino Nano.

ຕາຕະລາງ 4.3
ລາຍລະອຽດ PIN ຂອງ Arduino NANO

PIN	ລາຍລະອຽດ
ວິນ	ແຮງດັນພາຍນອກກັບກະດານ Output ເປັນ
+ 5 ວ	+5 V
3.3 ວ	ການສະຫນອງ 3.3 V ຢູ່ເທິງຫ
GND	ນໍ້າດິນ
IOREF	ມັນຊ່ວຍເລືອກແຫຼ່ງພະລັງງານທີ່ເໝາະສົມໂດຍການສະຫນອງ a ການອ້າງອີງແຮງດັນ
ລໍາດັບ ການຂັດຂວາງພາຍນອກ	ມັນ ສາ ມາດ ສົ່ງ ແລະ ຮັບ ຂໍ້ ມູນ serial ດ້ວຍ 0(Rx) ແລະ 1(Tx) ສົ່ງ ຜົນ ໃຫ້ ເກີດ ການ ຂັດ ຂວາງ ໃນ ຄ່າ ຕໍ່ າ (pins 2 ແລະ 3)
PWM	8 ບິດ PWM (3, 5, 6, 9, 10, 11)
SPI	ມັນສະຫນັບສະຫນູນການສື່ສານ SPI ກັບ [10 (SS), 11 (MOSI), 12 (MISO) ແລະ 13 (SCK)]
LED	LED ຂັບເຄື່ອນຢູ່ທີ່ pin 13
I2C	ຮອງຮັບສອງສາຍເຊື່ອມຕໍ່ [A4 (SDA) ແລະ A5 (SCL)] ມັນເປັນແຮງດັນອ້າງ
AREF	ອີງສໍາລັບການປ້ອນຂໍ້ມູນການປຸງປຸງທຽບ
ຮີເຊັດ	ມັນຖືກນໍາໃຊ້ເພື່ອປັບ microcontroller ໃນກະດານ

ຕາຕະລາງ 4.4

ຕາຕະລາງປຽບທຽບສໍາລັບກະດານ Arduino ຈໍານວນຫນ້ອຍ

ຊື່	ໂຮງງານຜະລິດ	CPU ຄວາມໄວ	ປະຕິບັດການ /	ດິຈິຕອລ IO/ PWM	ອະນາລັອກ ເຂົ້າ/ອອກ	UART	Flash [kB]
			ປ່ອນຂໍ້ມູນ ແຮງດັນ				
LilyPad	ATmega168V ATmega328P	8 MHz	2.4–5.5 V/ 2.4–5.5 V	14/6	6/0	—	16
ເມກາ 2560 ຈຸນລະພາກ	ATmega2560	16 MHz	5 V/4–12 V	54/15	16/0	4	256
	ATmega32U4	16 MHz	5 V/4–12 V	20/4	12/0	1	32
ຢູໂນ	ATmega328P	16 MHz	5 V/4–12 V	14/6	6/0	1	32
ລີໂອນາໂດ	ATmega32U4	16 MHz	5 V/4–12 V	20/4	12/0	1	32
ຢູນ	ATmega32U4	16 MHz	5 ວ	20/4	12/0	1	32
	AR9331 Linux	400 MHz					
ອິເທີເນັດ	ATmega328P	16 MHz	5 V/4–12 V	14/4	6/0	—	32
Gemma	ATtiny85	8 MHz	3.3 V/ 4–16 ວ	3/2	1/0	—	8
MKRZero	SAMD21 Cortex-M0+ 32 ບິດຕົ້າ ພະລັງງານ ARM MCU	48 MHz	3.3 ວ	22/12	4 (ADC 8/10/ 12 ບິດ)/1 (DAC 10 ບິດ)	1	256

4.2 Arduino IDE

ສະພາບແວດລ້ອມການພັດທະນາປະສົມປະສານ Arduino (IDE) ເປັນຊອບແວທີ່ເປີດ, ແລະເຮັດໃຫ້ມັນງ່າຍຕໍ່ການຂຽນລະຫັດແລະອັບໂຫລດມັນໃສ່ກະດານ.

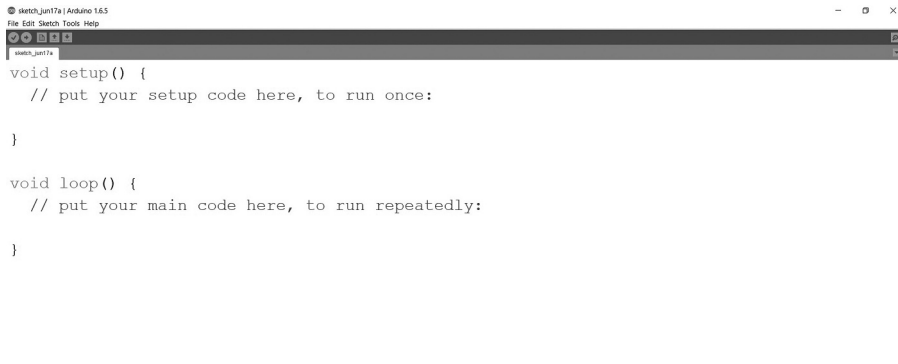
4.2.1 ຂັ້ນຕອນການຕິດຕັ້ງ Arduino IDE

ຂັ້ນຕອນທີ 1: ຕິດຕັ້ງ Arduino IDE ແລະເປີດປ້ອງຢ້ຽມ

ເພື່ອເລີ່ມຕົ້ນ, ຕິດຕັ້ງ Arduino IDE. [ຮູບທີ 4.4](#) ສະແດງໃຫ້ເຫັນປ້ອງຢ້ຽມຂອງ Arduino IDE.

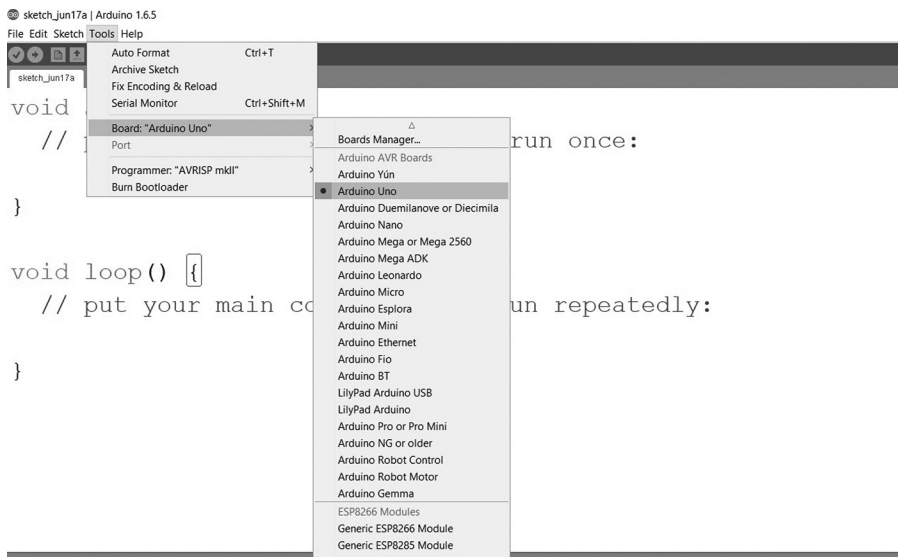
ຂັ້ນຕອນທີ 2: ເລືອກສະບັບຂອງກະດານ Arduino

Arduino ມີຫຼາຍລຸ້ນເຊັ່ນ UNO, MEGA, NANO, ແລະອື່ນໆ. ໃນໂຄງການ, ຊອກຫາສະບັບທີ່ເໝາະສົມໂດຍການເລືອກພາລາມີເຕີຕາມຄວາມຕ້ອງການ. ກະດານທົ່ວໄປທີ່ສຸດສໍາລັບຜູ້ເລີ່ມຕົ້ນແມ່ນ Arduino UNO. ເລືອກກະດານແລະພອດ serial



ຮູບທີ 4.4

Window Arduino IDE.



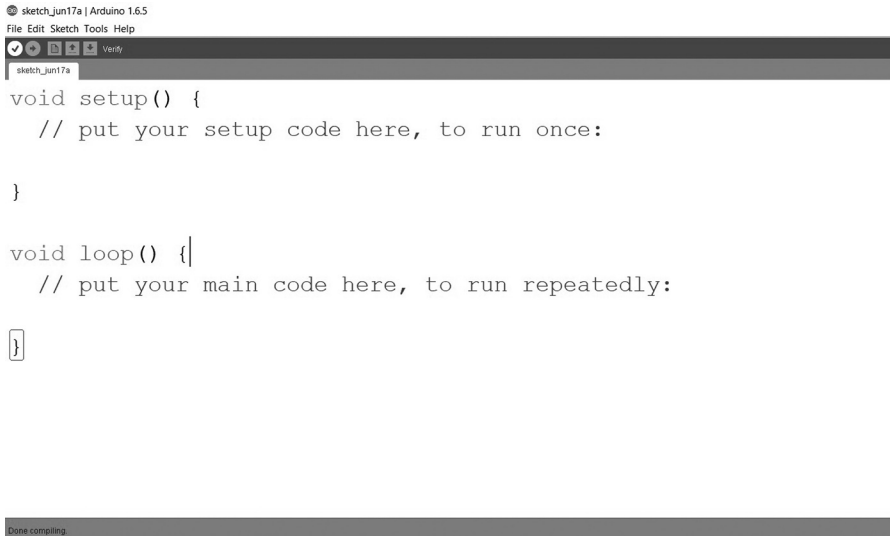
ຮູບທີ 4.5

ການເລືອກກະດານ Arduino.

ໃນ Arduino IDE. ເພື່ອເລືອກກະດານ Arduino, ໃຫ້ຄລິກໃສ່ "ເຄື່ອງມື" ແລະຫຼັງຈາກນັ້ນ ໃຫ້ຄລິກໃສ່ "ກະດານ." **ຮູບທີ 4.5**ສະແດງໃຫ້ເຫັນການເລືອກ "Arduino Uno."

ຂັ້ນຕອນທີ 3: ຂຽນແລະລວບລວມໂຄງການ

ຂຽນໂຄງການຢູ່ໃນປ່ອງຢ້ຽມ Arduino IDE. ຫຼັງຈາກນັ້ນ, "RUN" ປະກອບ. **ຮູບທີ 4.6**ສະແດງໃຫ້ເຫັນປ່ອງຢ້ຽມທີ່ຈະລວບລວມໂຄງການ.



ຮູບທີ 4.6

ລວບລວມໂຄງການ.

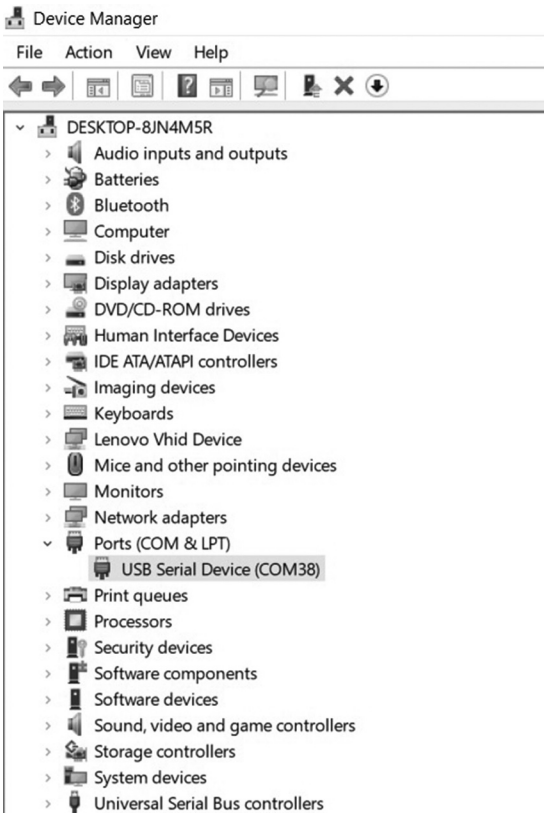
ຂັ້ນຕອນທີ 4: ເຊື່ອມຕໍ່ Arduino ກັບ PC

ເຊື່ອມຕໍ່ Arduino ກັບພອດ USB ຂອງ PC ດ້ວຍສາຍ USB. ທຸກໆ Arduino board ມີທີ່ຢູ່ serial-port ທີ່ແຕກຕ່າງກັນ (COM2, COM4, ແລະອື່ນໆ), ດັ່ງນັ້ນມັນຈຳເປັນຕ້ອງປັບຄ່າພອດສໍາລັບແຕ່ລະ Arduino ແລະເລືອກມັນໃນ IDE. ເພື່ອກວດເບິງພອດທີ່ Arduino ຖືກເຊື່ອມຕໍ່, ຄລິກຂວາໃສ່ "PC" ຈາກນັ້ນເລືອກ "ຜູ້ຈັດການ"; ປ່ອງຢ້ຽມຈະເປີດ. ຈາກນັ້ນກົດສອງຄັ້ງໃສ່ "Device Manager." ປ່ອງຢ້ຽມທີ່ສະແດງຢູ່ໃນ [ຮູບທີ 4.7](#) ຈະເປີດ. ໃຫ້ຄລິກໃສ່ພອດ COM ແລະ LPT ແລະພອດທີ່ອຸປະກອນເຊື່ອມຕໍ່ສາມາດພົບໄດ້.

ໃນບັດຈຸບັນໃຫ້ຄລິກໃສ່ "ເຄື່ອງມື" ຢູ່ໃນປ່ອງຢ້ຽມ Arduino IDE. ໄປທີ່ພອດແລະເລືອກເລກພອດດຽວກັນ, ເຊິ່ງພົບຢູ່ໃນຕົວຈັດການອຸປະກອນ (ເລືອກ COM1 ຫຼື COM2, ແລະອື່ນໆ). [ຮູບທີ 4.8](#) ສະແດງໃຫ້ເຫັນ "COM38" ເປັນພອດ serial ຂອງກະດານ.

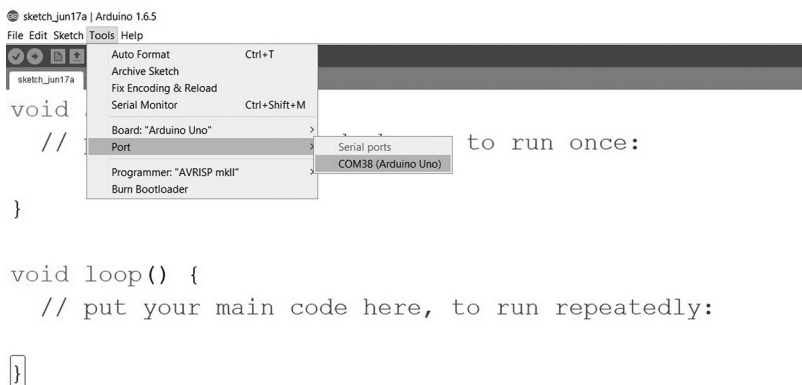
ຂັ້ນ ຕອນ ທີ 5 : ອັບ ໂຫລດ ໂຄງ ການ ກັບ Arduino board

ອັບໂຫລດໂປຣແກຣມໃສ່ກະດານ Arduino. [ຮູບທີ 4.9](#) ສະແດງ ໃຫ້ ເຫັນ ວິ ທີ ການ ອັບ ໂຫລດ ໂຄງ ການ .



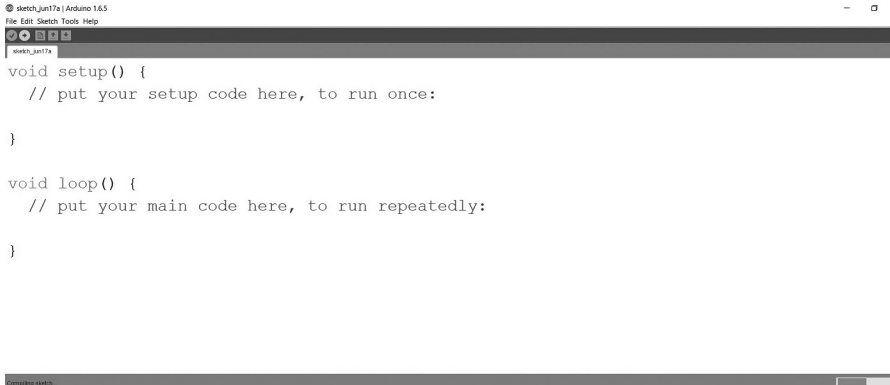
ຮູບທີ 4.7

ປ່ອງຢ້ຽມເພື່ອກວດສອບພອດຂອງ Arduino.



ຮູບທີ 4.8

ພອດ serial ຂອງຄະນະ.



ຮູບທີ 4.9

ປ້ອງຢ້ຽມເພື່ອອັບໂຫລດໂຄງການ.

4.3 ຄໍາສັ່ງພື້ນຖານສໍາລັບ Arduino

1. **pinMode(x, OUTPUT);**//ມອບຫມາຍເລກ PIN x ເປັນຂາອອກທີ x ເປັນຕົວເລກຂອງ PIN ດິຈິຕອນ
2. **digitalWrite(x, HIGH);**//ເປີດເລກ PIN x ເປັນ HIGH ຫຼື ON ບ່ອນທີ x ເປັນຕົວເລກຂອງ PIN ດິຈິຕອລ
3. **pinMode(x, INPUT);**//ມອບຫມາຍເລກ PIN x ເປັນ PIN ປ້ອນບ່ອນທີ x ເປັນຕົວເລກຂອງ PIN ດິຈິຕອນ
4. **digitalRead(Pin ດິຈິຕອລ);**//ອ່ານ pin ດິຈິຕອນເຊັ່ນ 13 ຫຼື 12 ຫຼື 11 ແລະອື່ນໆ.
5. **analogRead(pin analog);**//ອ່ານ pin analog ເຊັ່ນ A0 ຫຼື A1 ຫຼື A2 ແລະອື່ນໆ.

4.4 ຄໍາສັ່ງ LCD

1. **lcd.begin(16, 2);**//ເລີ່ມຕົ້ນ LCD 16*2 ຫຼື 20*4
2. **lcd.print("RAJESH");**//ພິມສາຍ "RAJESH" ໃນ LCD
3. **lcd.setCursor(x, y);**//ຕັ້ງຕົວກະພິບຂອງ LCD ໃນບ່ອນທີຕ້ອງການໂດຍທີ x ເປັນຈຳນວນ COLUMN ແລະ y
4. **lcd.print(LPU);**//ພິມ LPU ເປັນຈຳນວນເຕັມໃນ LCD
5. **lcd.Clear();**//ລຶບລ້າງເນື້ອໃນຂອງ LCD

4.5 ຄໍາສັ່ງການສື່ສານ Serial

- 1.**Serial.begin(baudrate);**//ເລີ່ມຕົ້ນການສື່ສານ serial ເພື່ອກຳນົດອັດຕາ baud ເປັນ 600/1200/2400/4800/9600
- 2.**Serial.print("RAJESH");**//serial print fixed string ທີ່ມີກຳນົດອັດຕາ baud ໃນແຖວ Tx
- 3.**Serial.println("RAJESH");**//serial print fixed string ກັບກຳນົດອັດຕາ baud ແລະໃສ່ຄໍາສັ່ງໃນແຖວ Tx
- 4.**Serial.print("LPU");**//serial print int string ທີ່ມີກຳນົດອັດຕາ baud ໃນແຖວ Tx
- 5.**Serial.print("LPU");**//serial print int string ກັບກຳນົດອັດຕາ baud ແລະໃສ່ຄໍາສັ່ງໃນແຖວ Tx
- 6.**Serial.Write(BYTE);**//serial ໂອນຫນຶ່ງ byte ໃນແຖວ Tx
- 7.**Serial.read();**//ອ່ານຫນຶ່ງ byte serial ຈາກ Rx line

4.6 ຫຼິ້ນກັບ LED ແລະ Arduino

ໄດໂອດປອຍແສງ (LED) ແມ່ນອຸປະກອນທີ່ສາມາດໃຊ້ເປັນຕົວຊີ້ບອກໄດ້. ໄຟ LED ມີສອງຂັ້ວ, anode ແລະ cathode. LEDs ມີຢູ່ໃນສີທີ່ແຕກຕ່າງກັນ.[ຮູບທີ 4.10](#)ສະແດງ LED.

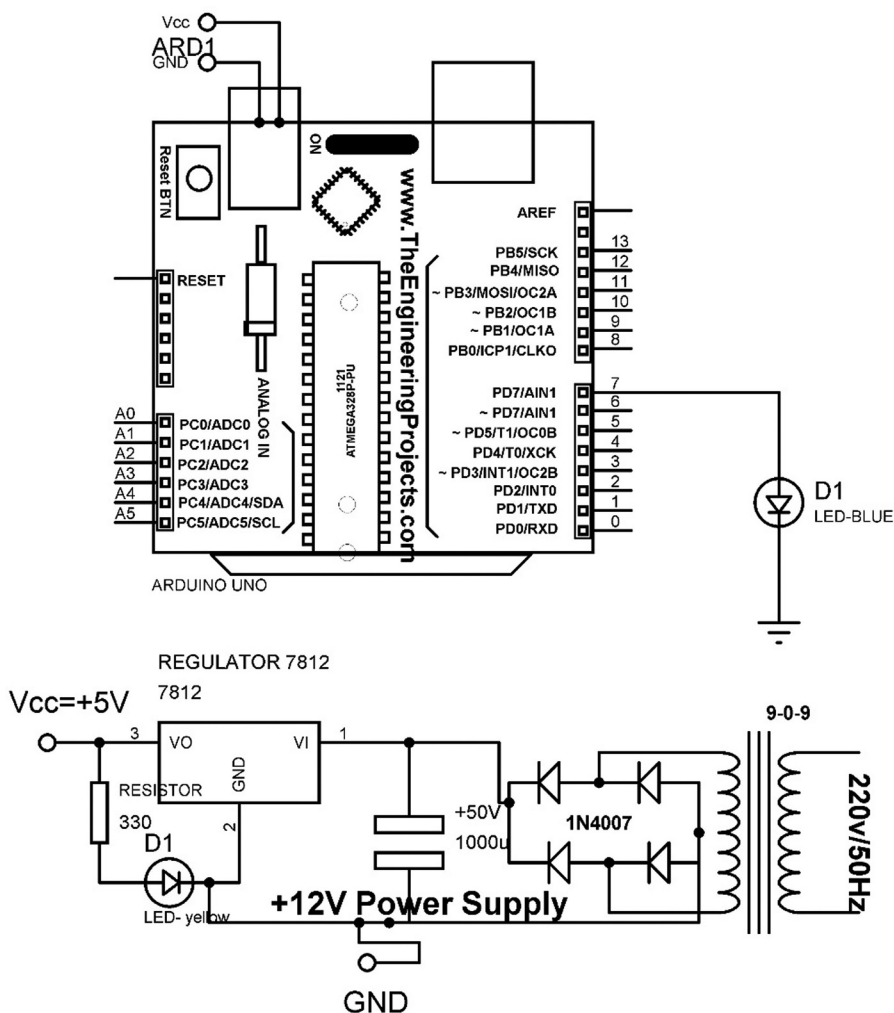


ຮູບ 4.10
ໄດໂອດປອຍແສງ.

ສີທີ່ແຕກຕ່າງກັນສາມາດຖືກນຳໃຊ້ເພື່ອເປັນຕົວແທນຂອງເງື່ອນໄຂທີ່ແຕກຕ່າງກັນ. ສີຂອງ LED ແມ່ນເນື່ອງມາຈາກການປ່ອຍແສງສະຫວ່າງໃນພາກພື້ນສະເພາະຂອງ spectrum ແສງສະຫວ່າງທີ່ສັງເກດເຫັນໂດຍທາດປະສົມທີ່ແຕກຕ່າງກັນ.

ເພື່ອເຂົ້າໃຈການເຮັດວຽກຂອງ LED, ເຊື່ອມຕໍ່ anode ຂອງ LED ກັບ pin 4 ຂອງ Arduino ແລະ cathode ກັບດິນ. ອັບໂຫຼດຮູບແຕ້ມທີ່ອະທິບາຍໄວ້ໃນພາກ 4.4.1 ໄປໃສ່ Arduino ແລະສັງເກດການກະພົບຂອງໄຟ LED.

ຮູບທີ 4.11 ສະແດງແຜນວາດວົງຈອນຂອງ Arduino interfacing ກັບ LED.



ຮູບທີ 4.11

ແຜນວາດວົງຈອນເພື່ອຕັດຕອບກັບ LED ກັບ Arduino.

4.6.1 ແຜນຜັງ

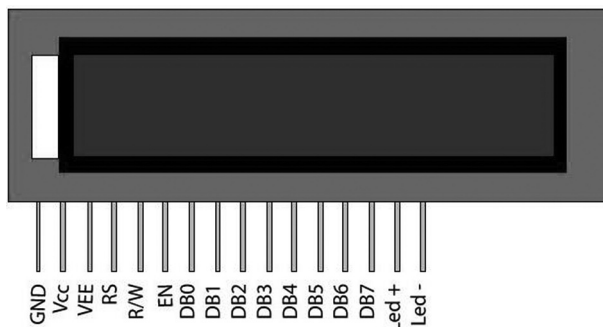
```
int LED_CONTROL=4;
ການຕັ້ງຄ່າ void()
{
    pinMode(LED_CONTROL, OUTPUT); // initialize pin 4 as output pin
}
void loop()
{
    digitalWrite(LED_CONTROL, HIGH); // ເຮັດໃຫ້ pin 4 HIGH
    delay(1000); // ຄວາມລ່າຊ້າ 1000 mS
    digitalWrite(LED_CONTROL, ຕົ້າ); // ເຮັດໃຫ້ pin 4 HIGH
    delay(1000); // ຄວາມລ່າຊ້າ 1000 mS
}
```

4.7 ຫຼິ້ນກັບ LCD ກັບ Arduino

ຈໍສະແດງຜົນໄປເຊຍກັນເປັນຂອງແຫຼວ (LCD) ແມ່ນໂມດູນຈໍສະແດງຜົນທີ່ໃຊ້ທົ່ວໄປ. A 16×2 ຈໍ LCD ຖືກນໍາໃຊ້ເປັນອຸປະກອນສະແດງຢູ່ໃນວົງຈອນ. ໂມດູນນີ້ແມ່ນຕ້ອງການຫຼາຍກວ່າເຈັດສ່ວນເພາະວ່າພວກເຂົາບໍ່ມີຂໍ້ຈຳກັດໃນການສະແດງພິເສດ, ແລະແມ້ກະທັ້ງ custom, ຕົວອັກສອນແລະປະຫຍັດ.

A 16×2 LCD ສາມາດສະແດງ 16 ຕົວອັກສອນຕໍ່ແຖວ, ແລະມີ 2 ແຖວ. ໃນ LCD ນີ້, 5×4 pixel matrix ສະແດງລັກສະນະ. ມັນມີສອງທະບຽນ, ຄື, ລົງທະບຽນຂໍ້ມູນແລະທະບຽນຄໍາສັ່ງ.[ຮູບທີ 4.12](#)ສະແດງໃຫ້ເຫັນ 16 ×2 ຈໍ LCD.

A 20×4 LCD ມີ 4 ແຖວແລະສາມາດສະແດງ 20 ຕົວອັກສອນຕໍ່ແຖວ. A 5×4 pixel matrix ຖືກນໍາໃຊ້ເພື່ອສະແດງຕົວອັກສອນ. ລາຍລະອຽດ PIN ແມ່ນຄືກັນກັບ LCD (16×2).[ຮູບທີ 4.13](#)ສະແດງໃຫ້ເຫັນ 20 ×4 ຈໍ LCD ([ຕາຕະລາງ 4.5](#)).



ຮູບທີ 4.12

ການ ສະ ແດງ ໄປ ເຊຍ ກັນ ຂອງ ແຫຼວ (16 ×2).



ຮູບທີ 4.13
ການ ສະ ແດງ ໄປ ເຊຍ ກັນ ຂອງ ແຫຼວ (20 ×4).

ຕາຕະລາງ 4.5

ລາຍລະອຽດ LCD Pin

ປັກໝຸດ	ລາຍລະອຽດ
Pin (1) ດິນ	ດິນ (0 V)
ພິນ (2) ວcc	ການສະຫນອງພະລັງງານ (5 V)
ພິນ (3) ວEE	ຕົວຕ້ານທານຕົວປ່ຽນແປງແມ່ນໃຊ້ເພື່ອປັບຄວາມຄົມຊັດ
Pin (4) ລົງທະບຽນ ເລືອກ	ເມື່ອຕົ້າ, ມັນເລືອກຄໍາສັ່ງລົງທະບຽນ, ແລະຖ້າສູງ, ຫຼັງຈາກນັ້ນມັນເລືອກ ການ ຈັດ ທະ ບຽນ ຂໍ້ ມູນ
ປັກໝຸດ (5) ອ່ານ/ຂຽນ	ສູງ ເພື່ອ ອ່ານ ບັນ ຊີ ແລະ ຕໍ່ າ ທີ່ ຈະ ຂຽນ ກ່ຽວ ກັບ ການ ຈັດ ທະ ບຽນ ສົ່ງ ຂໍ້ ມູນ ກັບ ເສັ້ນ ຂໍ້ ມູນ ໃນ ເວ ລາ ທີ່ ກໍາ ມະ ຈອນ ສູງ ຫາ ຕໍ່ າ ແມ່ນ ໄດ້ ຮັບ
Pin (6) ເປີດໃຊ້ງານ	
ປັກໝຸດ (7) DB0	
Pin (8) DB1	
Pin (9) DB2	ສາຍຂໍ້ມູນ 8 ບິດ
ປັກໝຸດ (10) DB3	
ປັກໝຸດ (11) DB4	
Pin (12) DB5	
Pin (13) DB6	
ປັກໝຸດ (14) DB7	
Pin (15) LED+	Backlight Vcc (5 V)
Pin (16) LED-	Backlight ground (0 V)

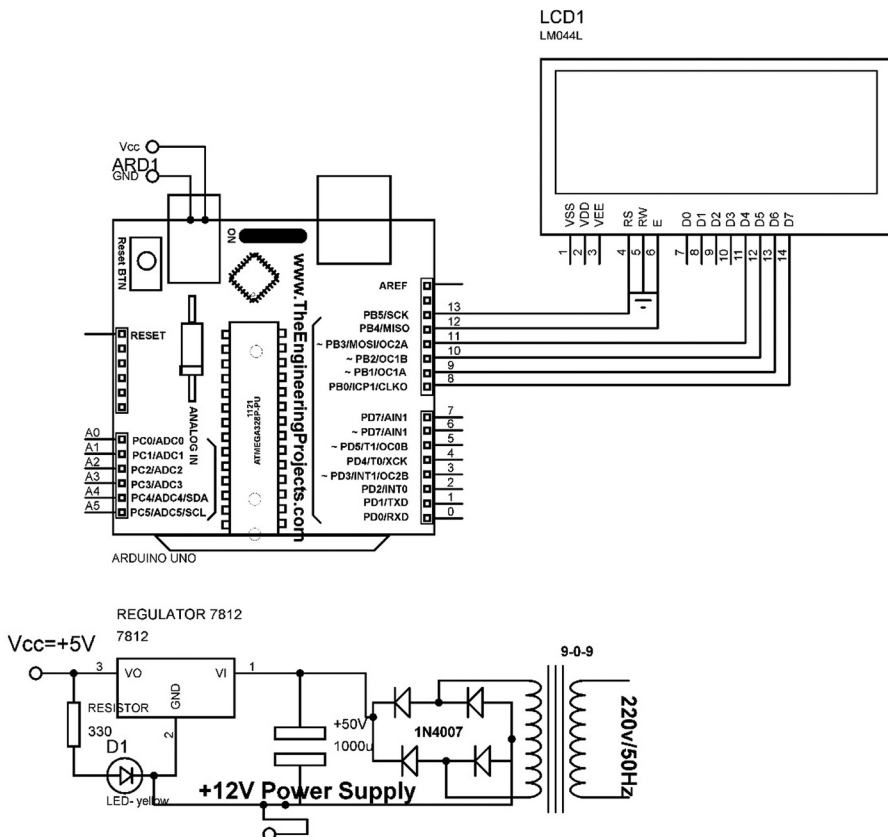
ການເຊື່ອມຕໍ່ LCD

ເຊື່ອມຕໍ່ອົງປະກອບດັ່ງຕໍ່ໄປນີ້:

- Arduino digital pin (13) ຫາ RS pin (4) ຂອງ LCD.
- Arduino digital pin (GND) ຫາ RW pin (5) ຂອງ LCD.
- Arduino digital pin (12) ຫາ E pin (6) ຂອງ LCD.

- Arduino pin digital (11) ຫາ D4 pin (11) ຂອງ LCD.
- Arduino pin digital (10) ຫາ D5 pin (12) ຂອງ LCD.
- Arduino ດິຈິຕອລ pin (9) ຫາ D6 pin (13) ຂອງ LCD.
- Arduino ດິຈິຕອລ pin (8) ຫາ D7 pin (14) ຂອງ LCD.

ຮູບທີ 4.14 ສະແດງແຜນວາດວົງຈອນຂອງ Arduino interfacing ກັບ LCD.



ຮູບທີ 4.14

ແຜນວາດວົງຈອນເພື່ອອ່ານ LCD.

4.7.1 ແຜນຜັງ

```
# ປະກອບມີ <LiquidCrystal.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

ການຕັ້ງຄ່າ void()
{
  lcd.begin(20, 4); // ເລີ່ມຕົ້ນ LCD lcd.print("ຍິນດີຕ້ອນ
  ຮັບ"); // Print string on LCD delay(2000); // Delay
  2000mS
  lcd.clear();
}

void loop()
{
  lcd.setCursor(0, 1); // ຕັ້ງຕົວກະພິບຂອງ LCD lcd.print("ECE
  Department"); // Print string on LCD delay(2000); //
  Delay 2000mS
  lcd.setCursor(0, 2); // ຕັ້ງຕົວກະພິບຂອງ LCD
  lcd.print("Rajesh Singh"); // Print string on LCD
  delay(2000); // Delay 2000mS
}
```