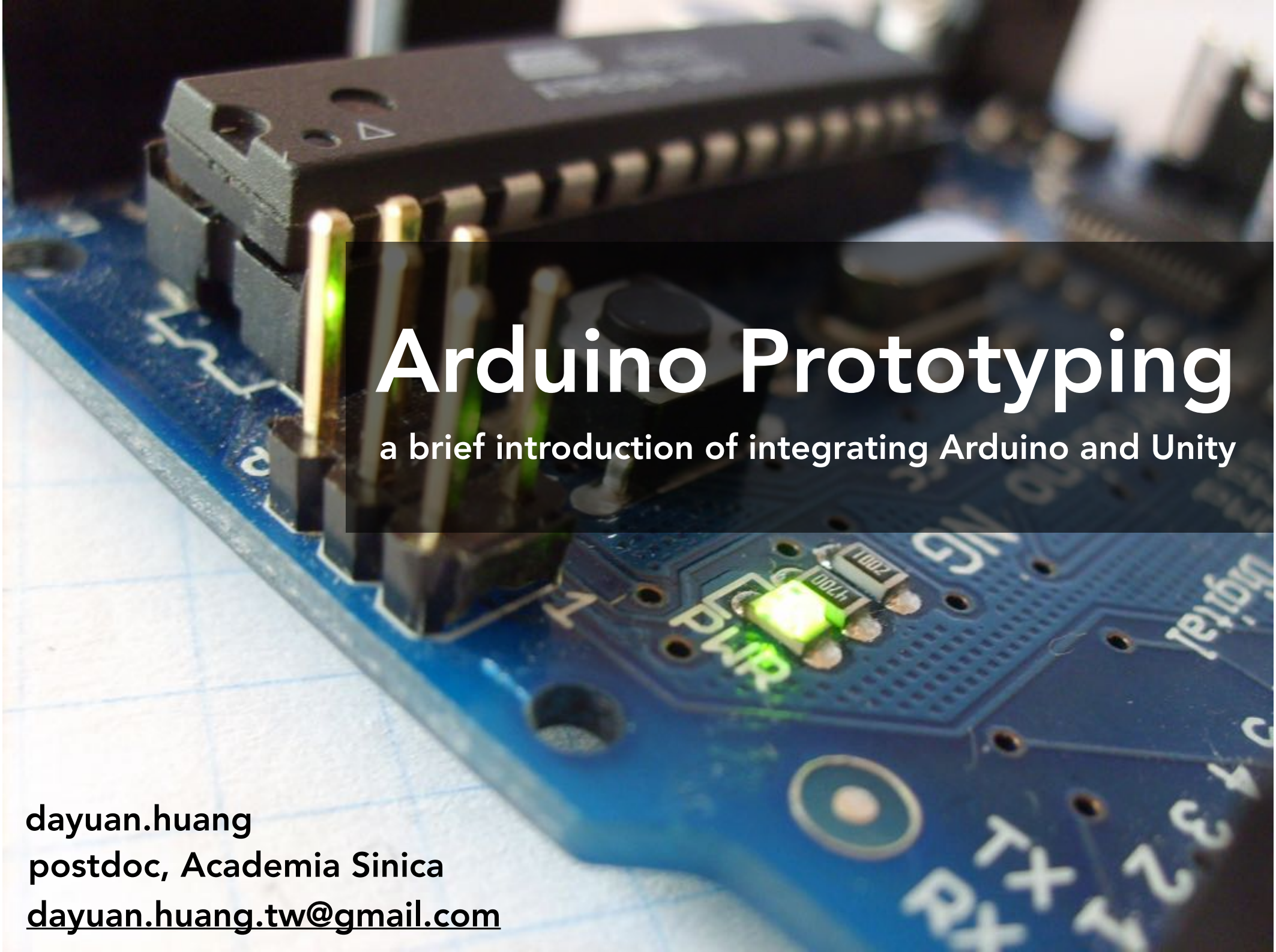
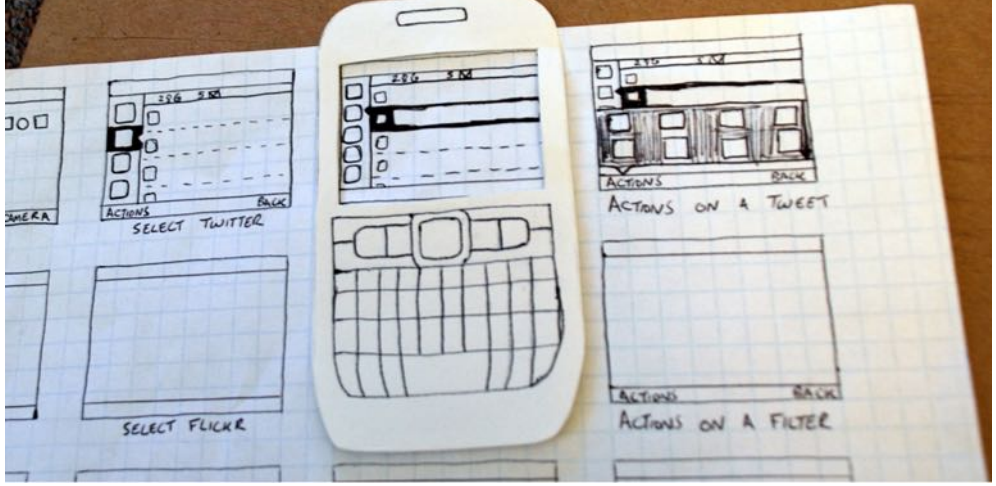


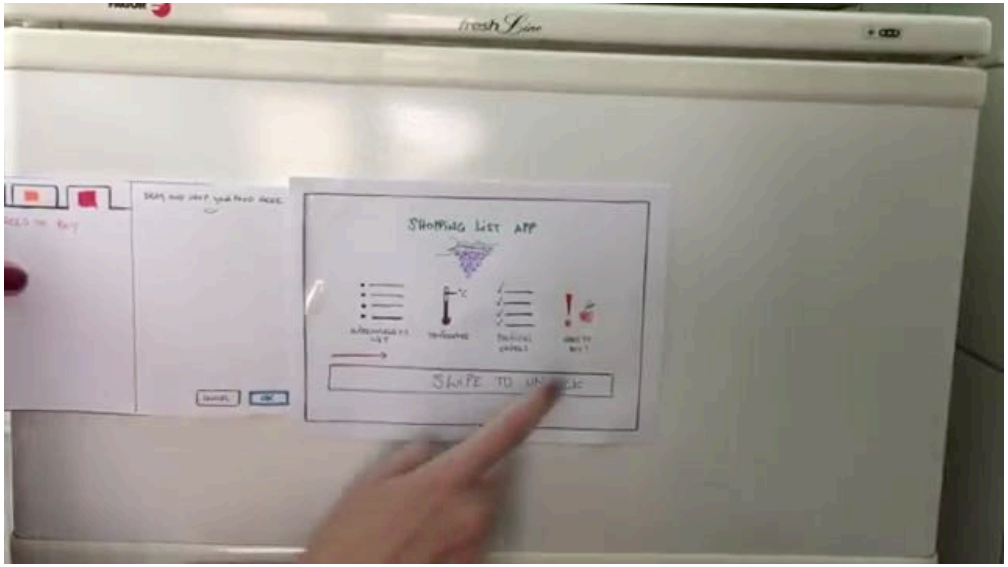
Arduino Prototyping

a brief introduction of integrating Arduino and Unity

dayuan.huang
postdoc, Academia Sinica
dayuan.huang.tw@gmail.com



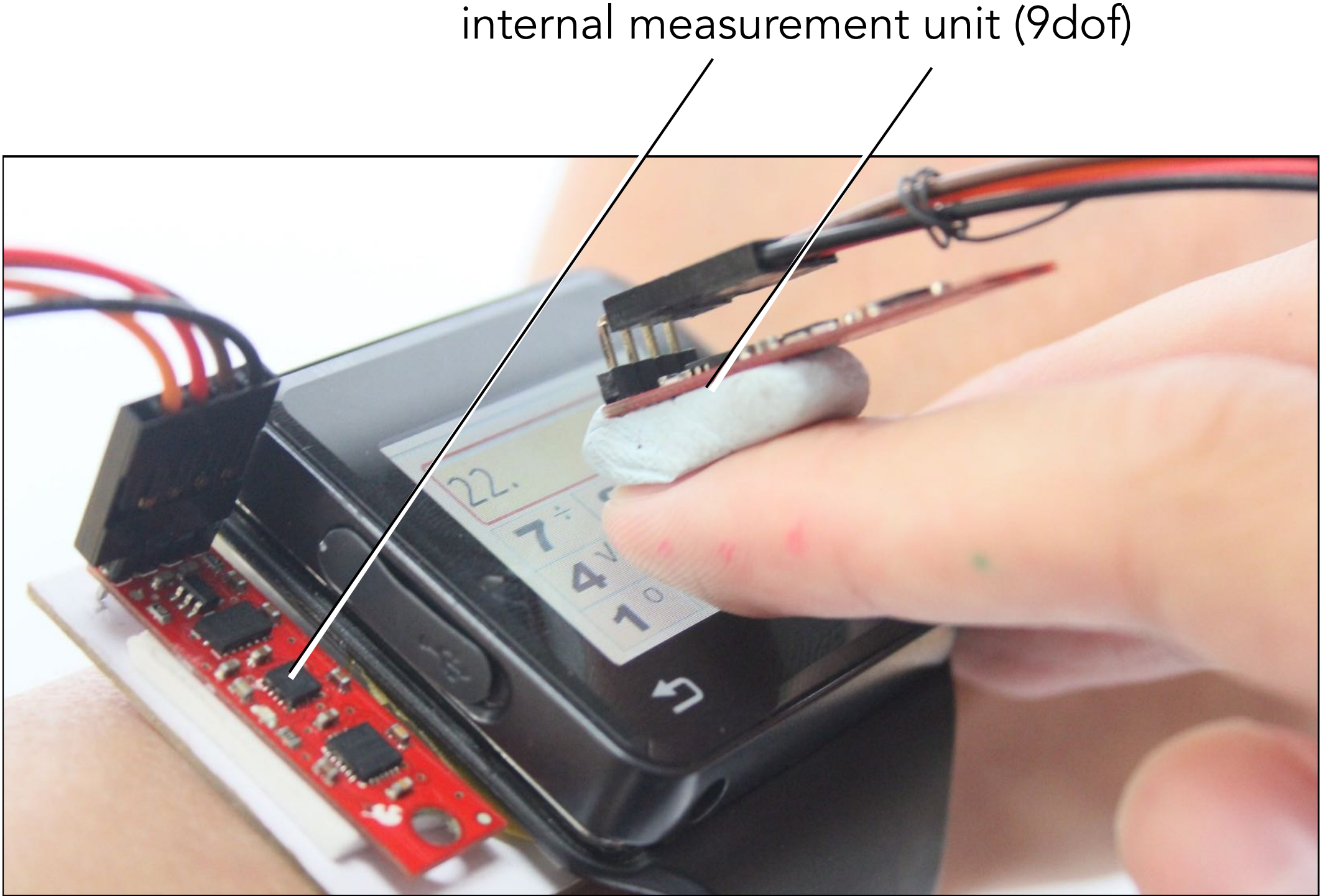
paper prototyping

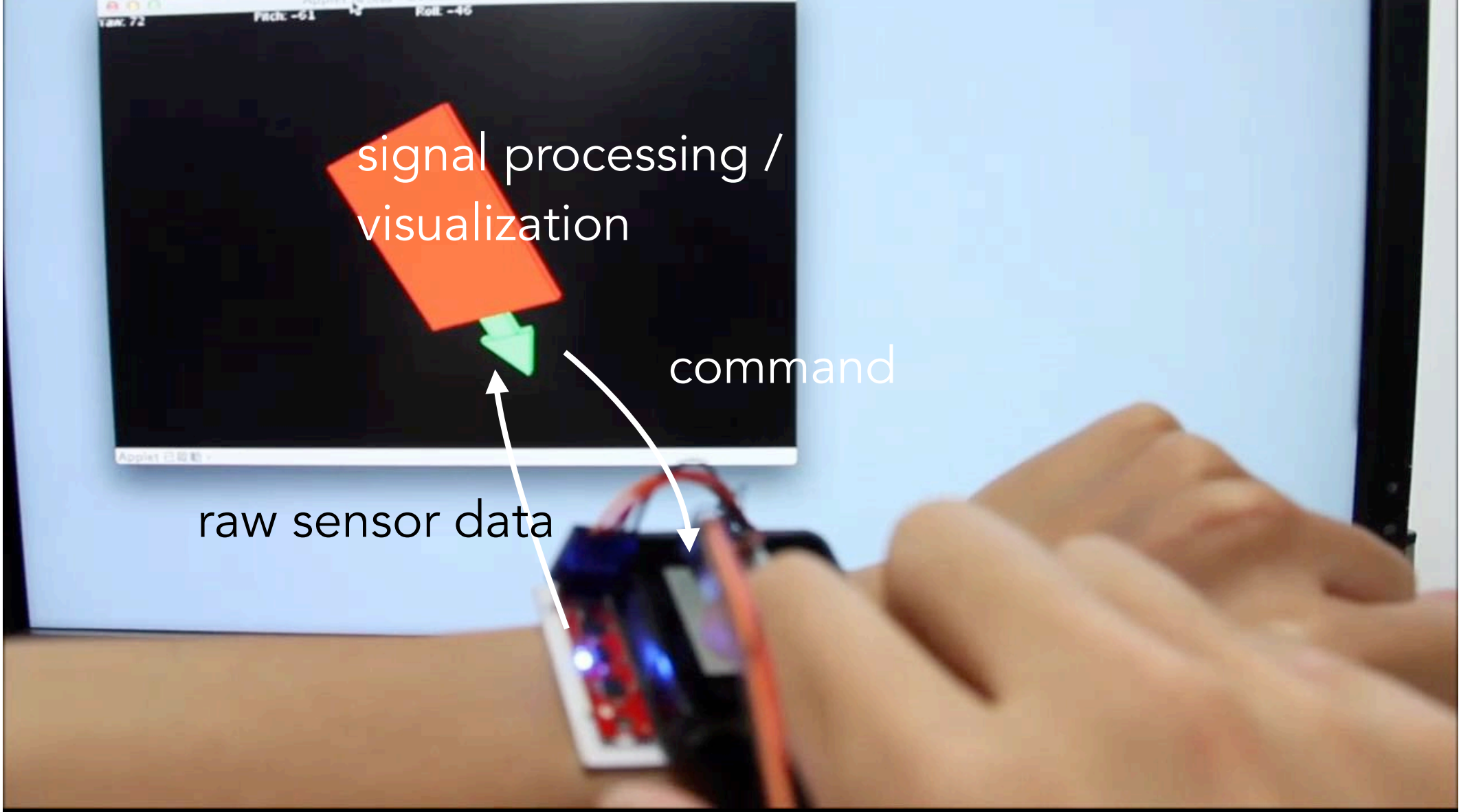


video prototyping

wizard of oz

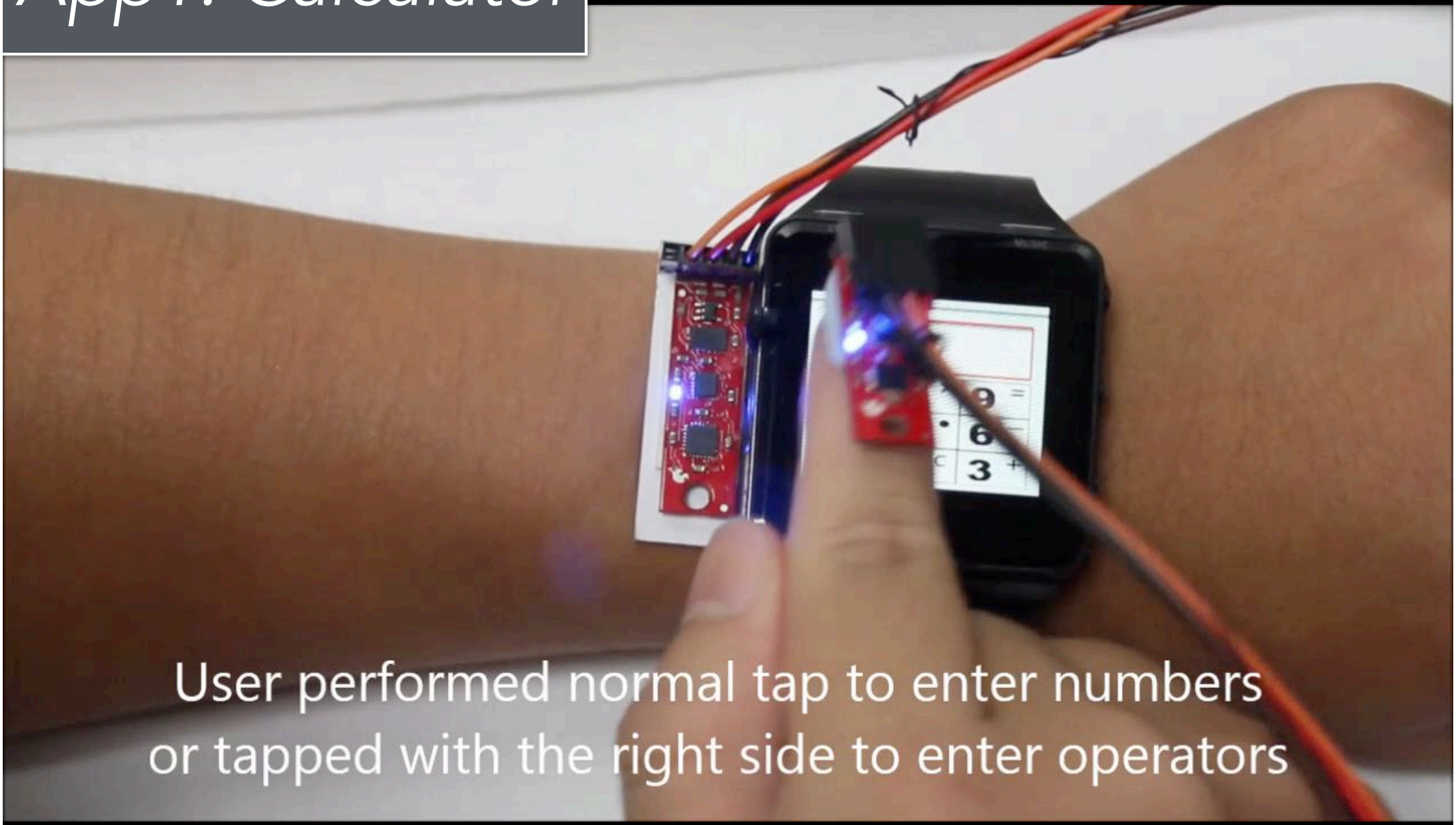
Hardware Prototype



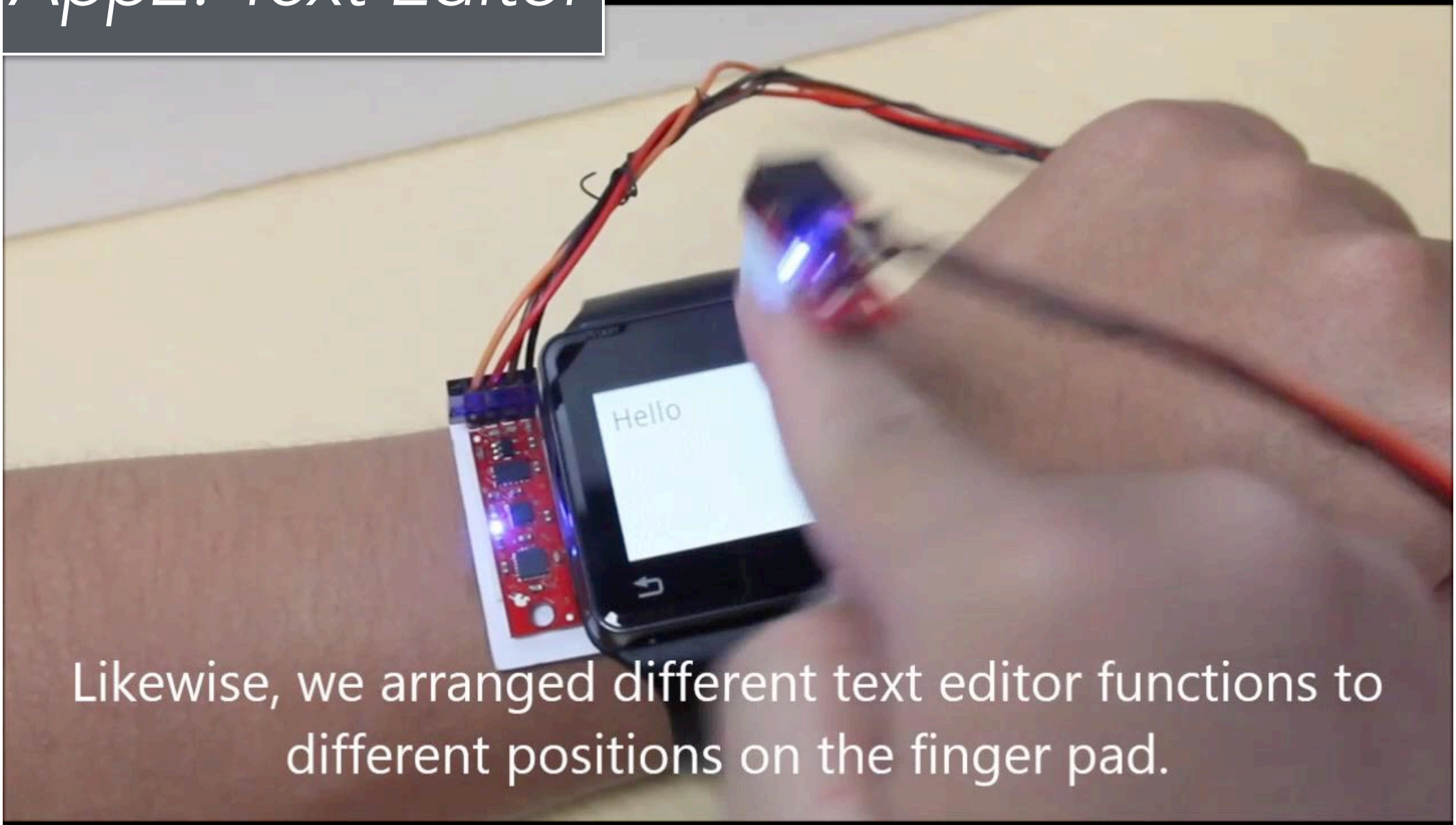


compute the orientation relationship between
the **touchscreen** and
the **index finger**

App1: Calculator



App2: Text Editor



Likewise, we arranged different text editor functions to different positions on the finger pad.

ARDUINO

http://arduino.cc/en/

1 Inexpensive

2 Cross-platform

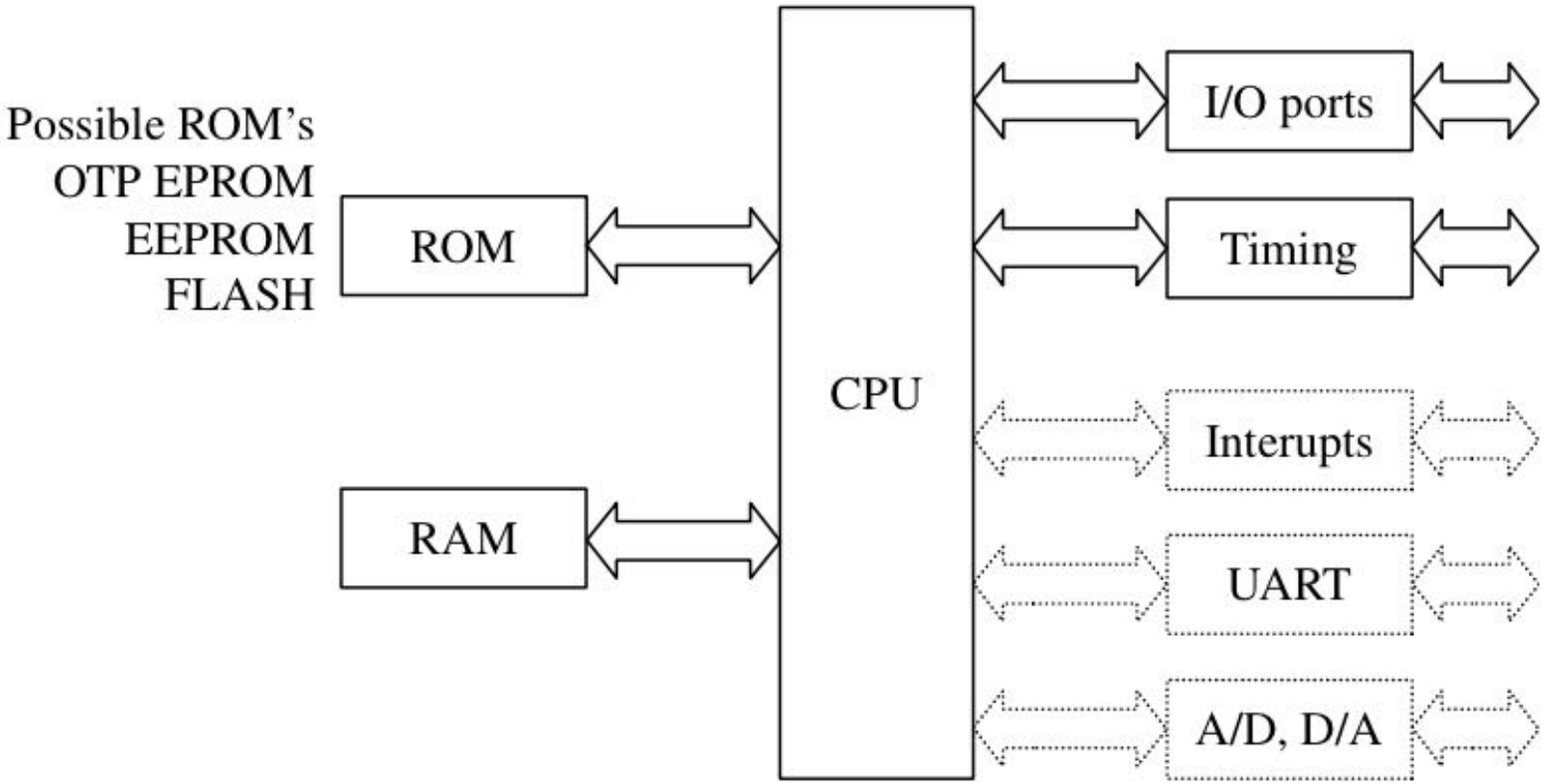
3 Open source and extensible software

4 Open source and extensible hardware

microcontroller a computer on a chip

CPU, ROM, RAM, Serial Communication Ports, etc.

microcontroller a computer on a chip



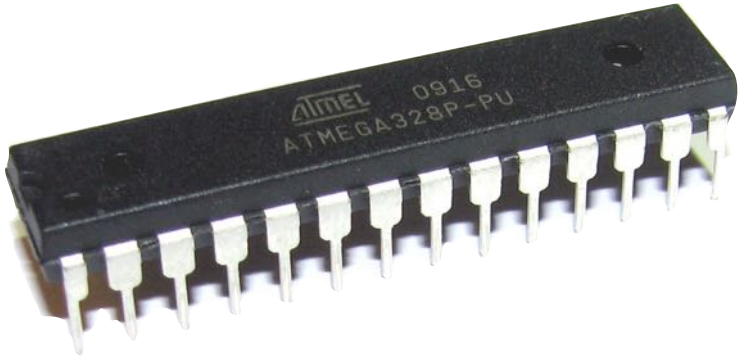
different microcontrollers

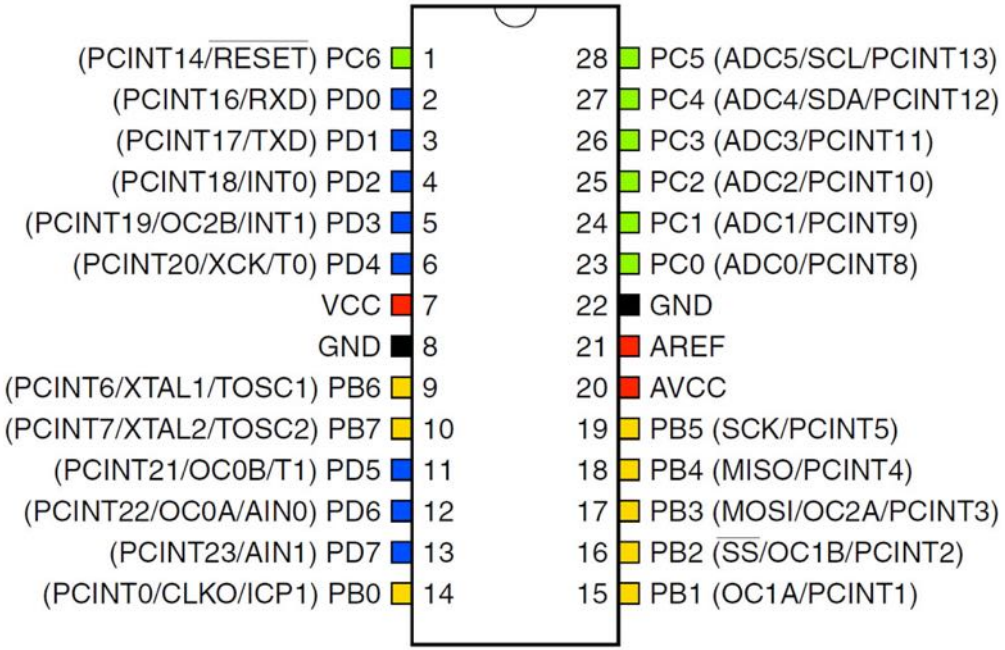
AtMega328

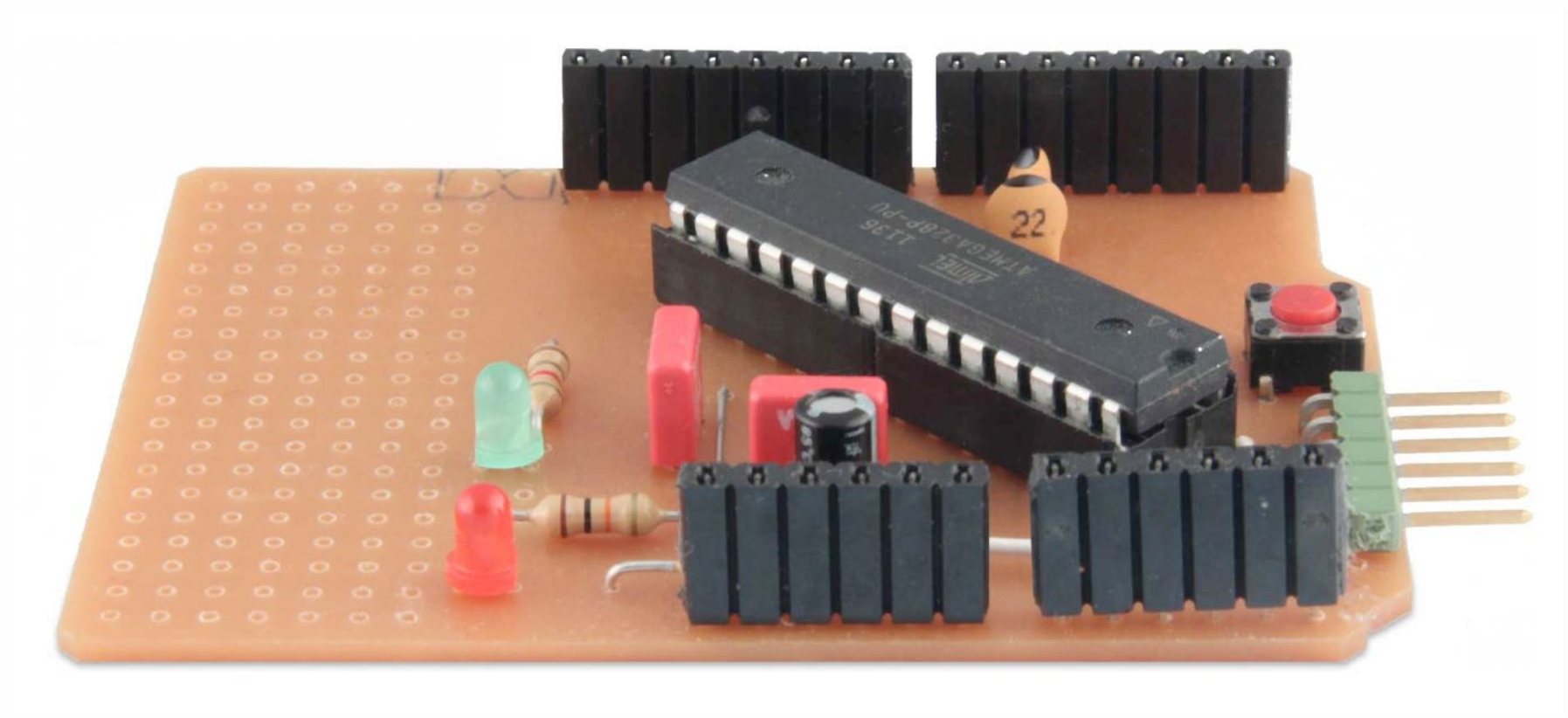
AtTiny58

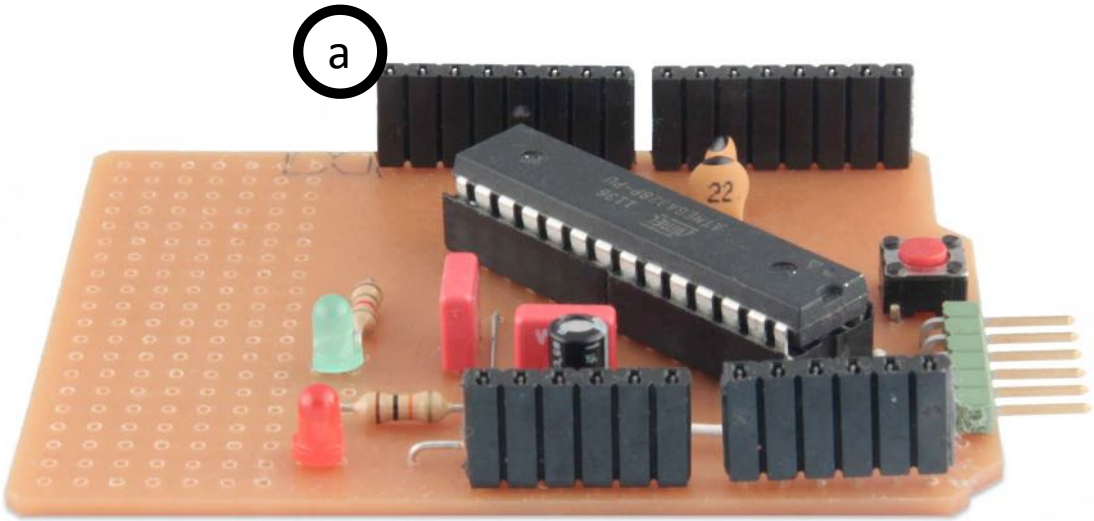
AtxMega128

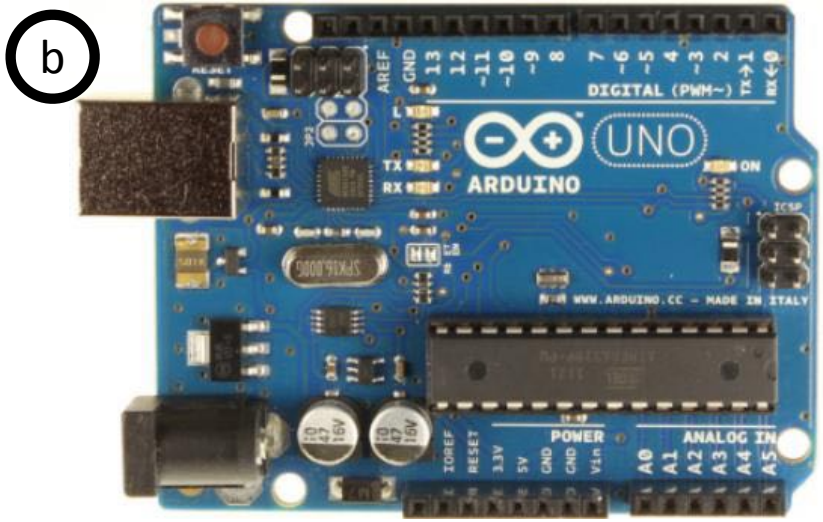
AtMega328

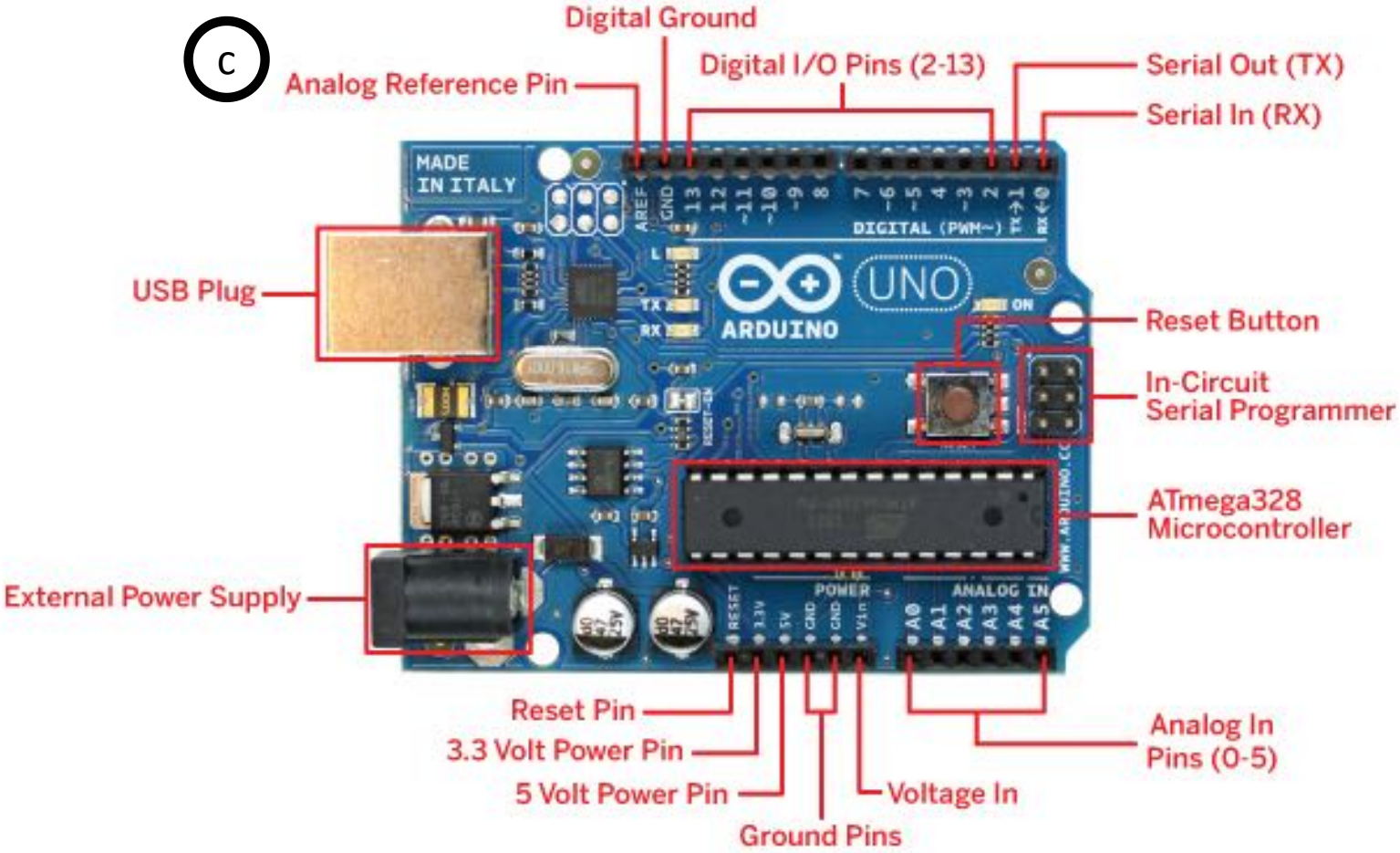

atmel
0916
ATMEGA328P-PU


(PCINT14/RESET) PC6 1
(PCINT16/RXD) PD0 2
(PCINT17/TXD) PD1 3
(PCINT18/INT0) PD2 4
(PCINT19/OC2B/INT1) PD3 5
(PCINT20/XCK/T0) PD4 6
VCC 7
GND 8
(PCINT6/XTAL1/TOSC1) PB6 9
(PCINT7/XTAL2/TOSC2) PB7 10
(PCINT21/OC0B/T1) PD5 11
(PCINT22/OC0A/AIN0) PD6 12
(PCINT23/AIN1) PD7 13
(PCINT0/CLKO/ICP1) PB0 14
28 PC5 (ADC5/SCL/PCINT13)
27 PC4 (ADC4/SDA/PCINT12)
26 PC3 (ADC3/PCINT11)
25 PC2 (ADC2/PCINT10)
24 PC1 (ADC1/PCINT9)
23 PC0 (ADC0/PCINT8)
22 GND
21 AREF
20 AVCC
19 PB5 (SCK/PCINT5)
18 PB4 (MISO/PCINT4)
17 PB3 (MOSI/OC2A/PCINT3)
16 PB2 (SS/OC1B/PCINT2)
15 PB1 (OC1A/PCINT1)


22

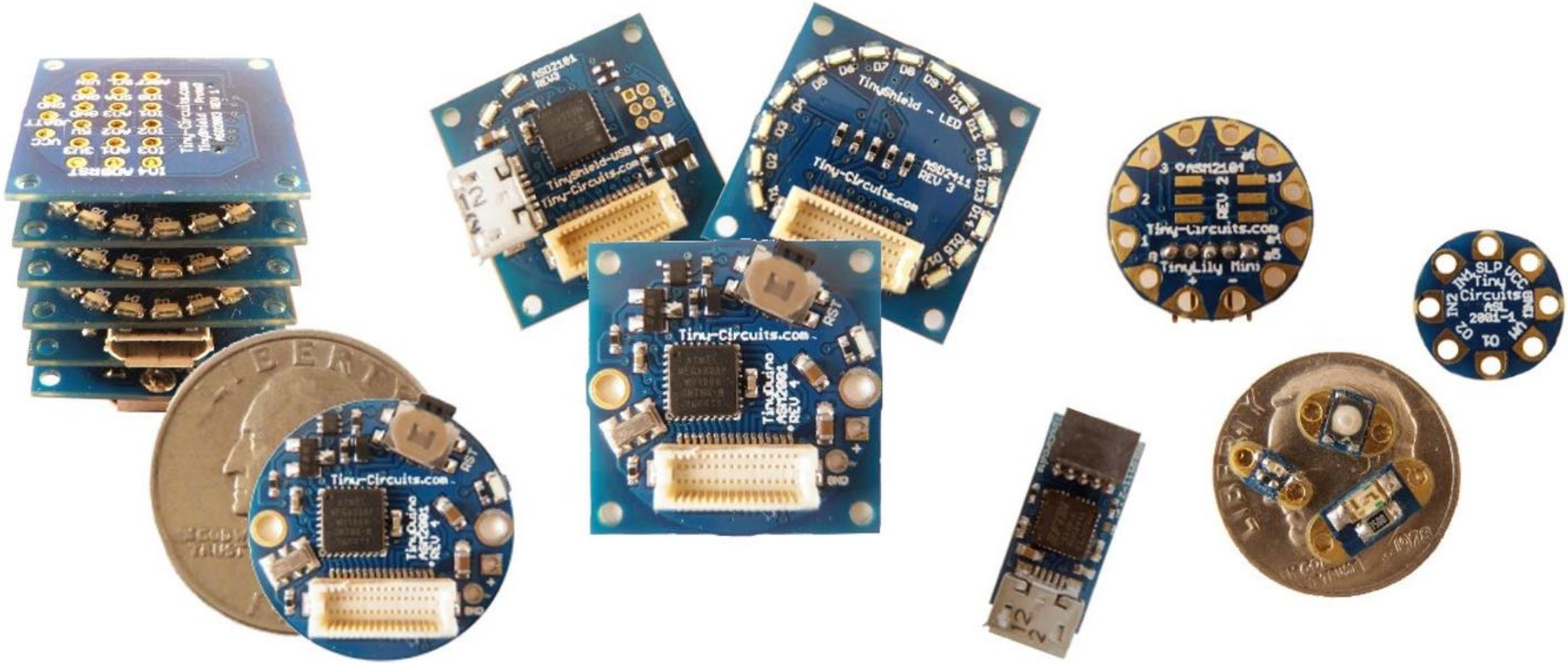

a


b
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

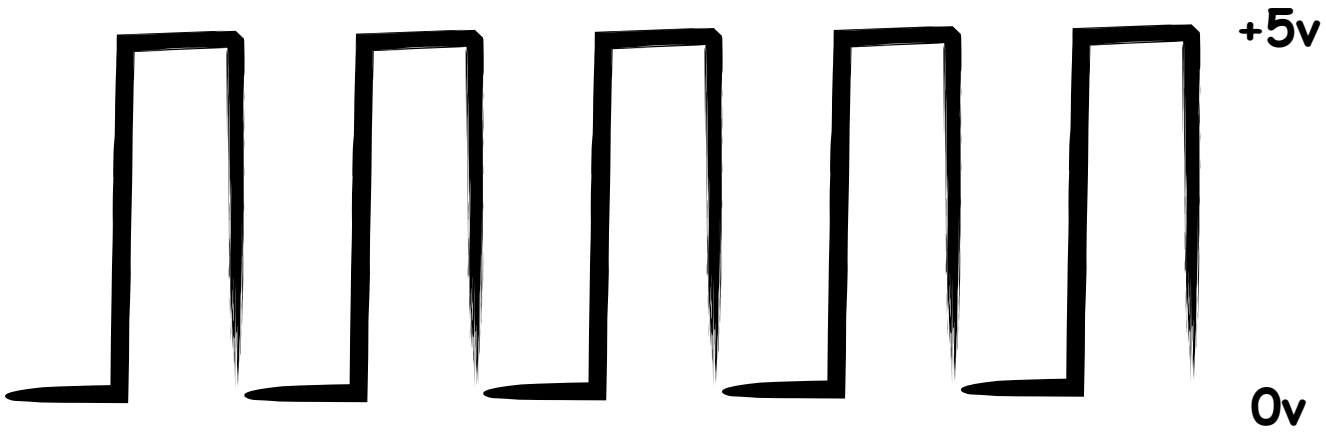

c
Digital Ground
Analog Reference Pin
Digital I/O Pins (2-13)
Serial Out (TX)
Serial In (RX)
USB Plug
Reset Button
In-Circuit
Serial Programmer
ATmega328
Microcontroller
External Power Supply
Reset Pin
3.3 Volt Power Pin
5 Volt Power Pin
Ground Pins
Voltage In
Analog In
Pins (0-5)
MADE
IN ITALY
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

TinyDuino

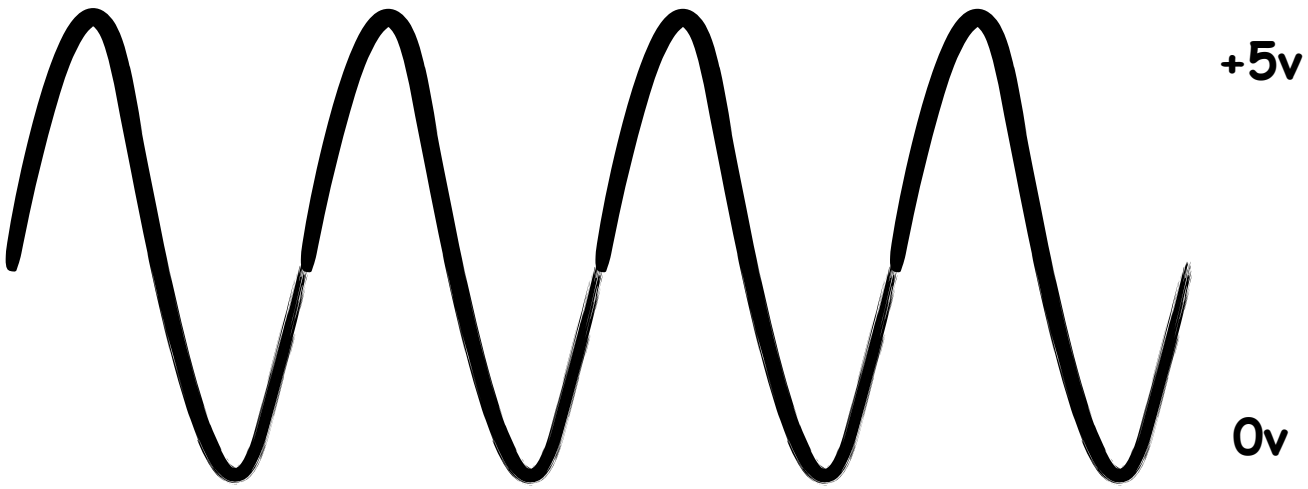
The Tiny Arduino Compatible Platform With Stackable Shield Support!



digital signal

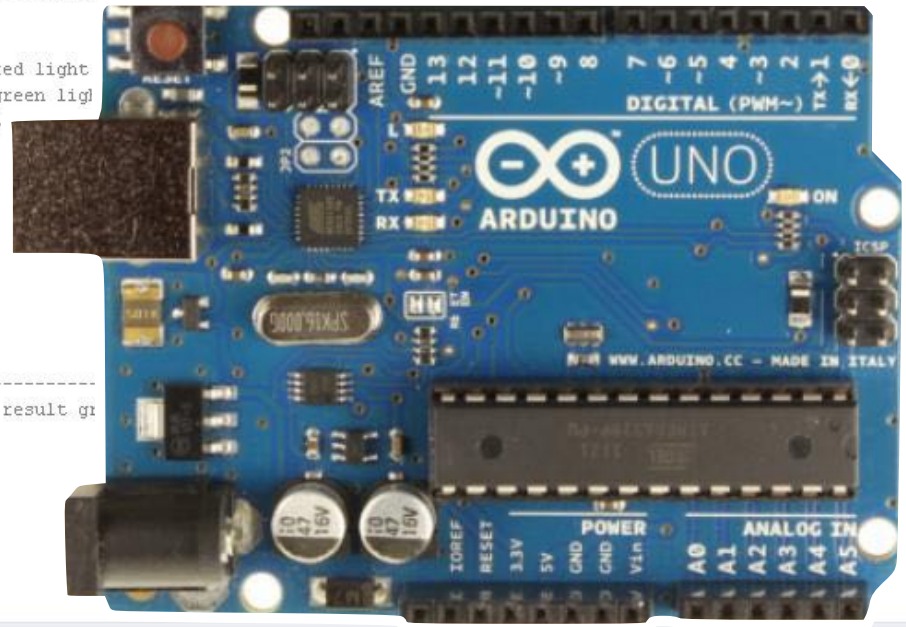


analog signal



Mr_Tidy_MkII_Sample_Code | ArmFunctions | CallibrationFunctions | CollisionAvoidance | ColourAction | FrontMidSensors | FrontSensors | GripperFunctions § | IOpindefinitions.h | Interrupts § | Notes | ObjectColour §

```
/=================================================== Determines the colour of an object ========================================
/                    NOTE: Due to the time this takes both robot and object need to be stationary for acurate readings.
/                                  Readings should be reasonably consistant regardless of ambient light.
void ObjectColour()

  analogWrite(RedLEDpin,0);                          // turn off red LED
  analogWrite(GreenLEDpin,0);                        // turn off green LED
  analogWrite(BlueLEDpin,0);                         // turn off blue LED
  delay(ldrdelay*2);                                 // allow time for LDR to respond
  ambientvalue=analogRead(RGBLDRpin);                // read ambient light

  analogWrite(RedLEDpin,pwmred);                     // turn on red LED to calibrated value
  delay(ldrdelay);                                   // allow time for LDR to respond
  redvalue=analogRead(RGBLDRpin);                    // read LDR with red LED on          (total light = ambient + reflected red light)
  analogWrite(RedLEDpin,0);                          // turn off red LED

  analogWrite(GreenLEDpin,pwmgreen);                 // turn on green LED to calibrated value
  delay(ldrdelay);                                   // allow time for LDR to respond
  greenvalue=analogRead(RGBLDRpin);                  // read LDR with green LED on        (total light = ambient + reflected green light)
  analogWrite(GreenLEDpin,0);                        // turn off green LED

  analogWrite(BlueLEDpin,pwmblue);                   // turn on blue LED to calibrated value
  delay(ldrdelay);                                   // allow time for LDR to respond
  bluevalue=analogRead(RGBLDRpin);                   // read LDR with blue LED on         (total light = ambient + reflected blue light)
  analogWrite(BlueLEDpin,0);                         // turn off blue LED

  redvalue=redvalue-ambientvalue;                    // calculate reflected red light     (reflected red light
  greenvalue=greenvalue-ambientvalue;                // calculate reflected green light   (reflected green lig
  bluevalue=bluevalue-ambientvalue;                  // calculate reflected blue light    (reflected b

  if(redvalue<0) redvalue=0;
  if(greenvalue<0) greenvalue=0;
  if(bluevalue<0) bluevalue=0;

  lightvalue=greenvalue;                             // determine minimum reading
  if (redvalue<greenvalue && redvalue<bluevalue) lightvalue=redvalue;
  if (bluevalue<greenvalue && bluevalue<redvalue) lightvalue=bluevalue;

  //-------------------------------------------------------------- Determine object colour----------------------------------
  //      Note: uses percentage of average value so that distance / brightness of the object does not affect result gr
  objectcolour=0;                                    // start with a value of 0 (nothing)

  if (lightvalue>(1000-ambientvalue)/8)
  {
    avr=redvalue*33/lightvalue*3;
    if (avr>coloursensitivity) objectcolour+=4;      // if red   is at least 14% greater than average
    avg=greenvalue*33/lightvalue*3;
    if (avg>coloursensitivity) objectcolour+=2;      // if green is at least 14% greater than average
```



Done Saving.

```
Binary sketch size: 23792 bytes (of a 126976 byte maximum)
```

57

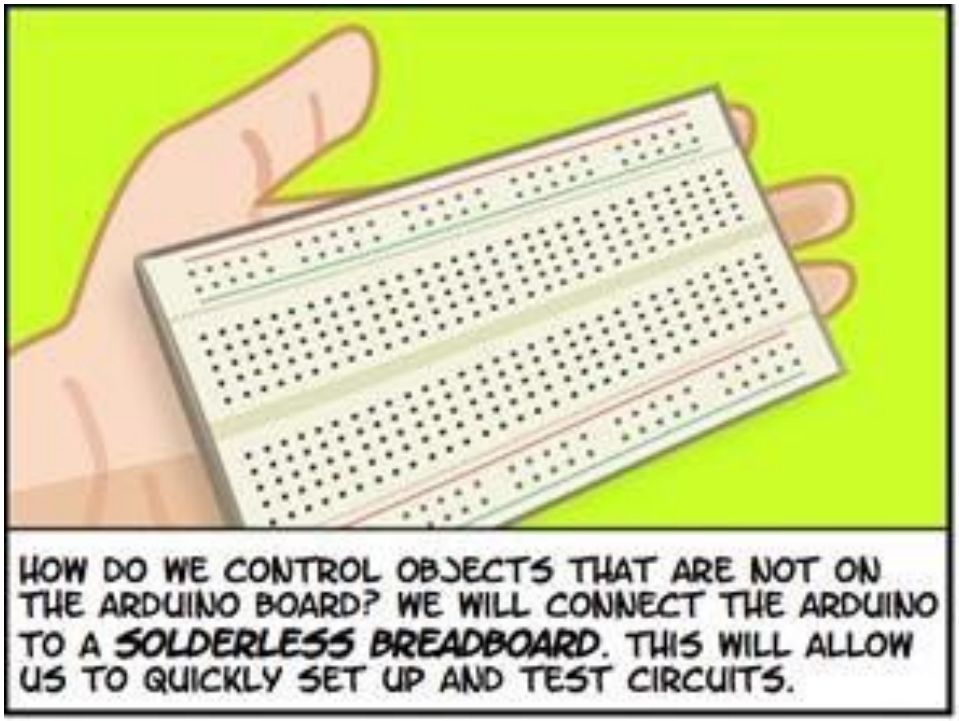
a buttons, pwm, and functions

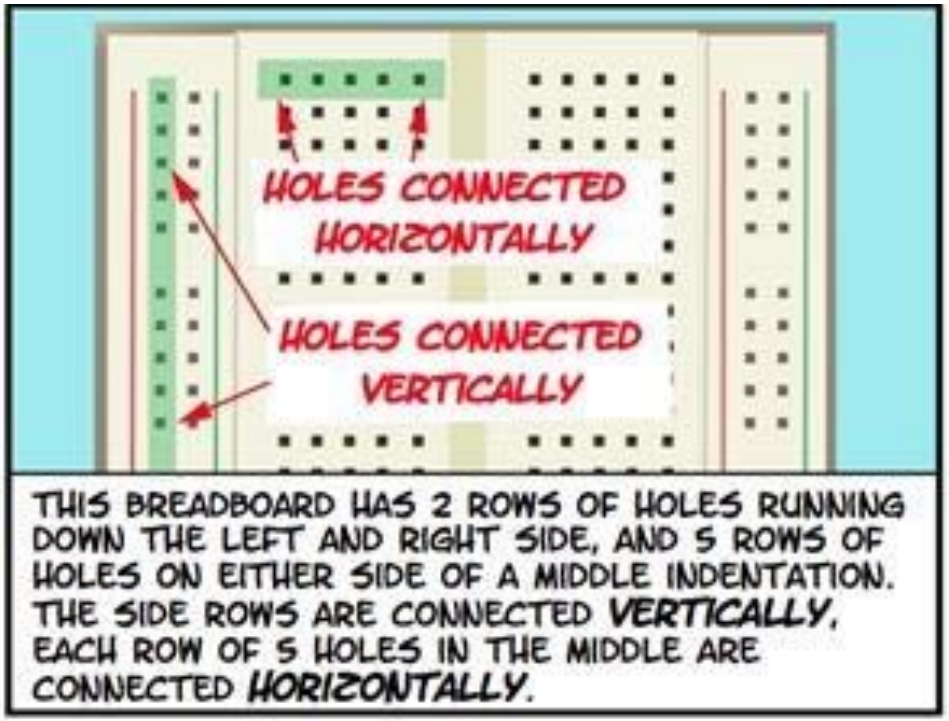
b electrical engineering

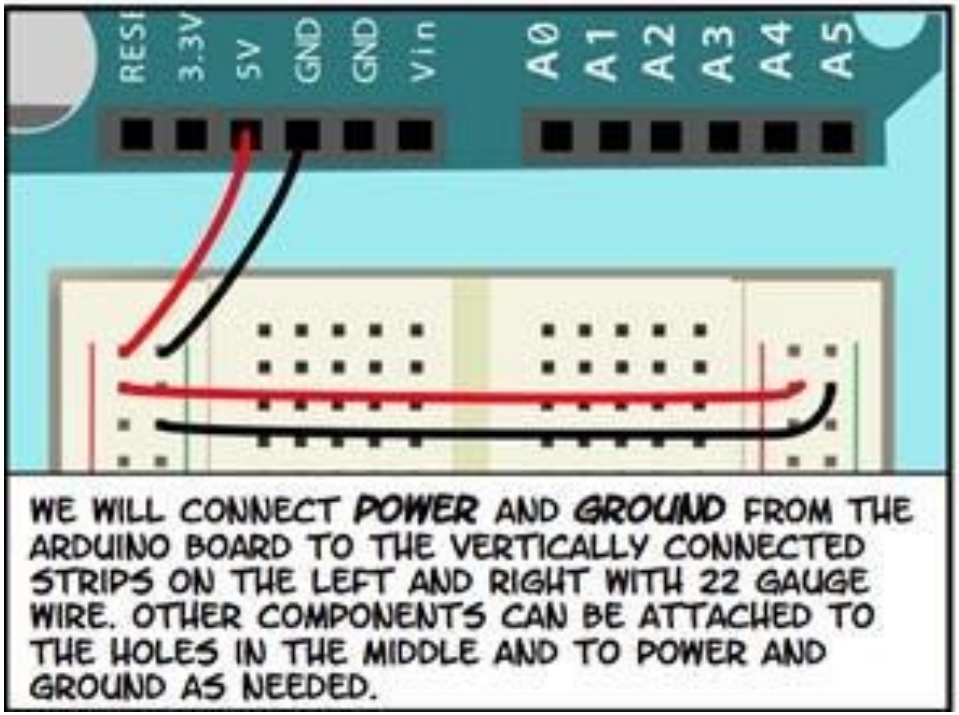
c analog inputs

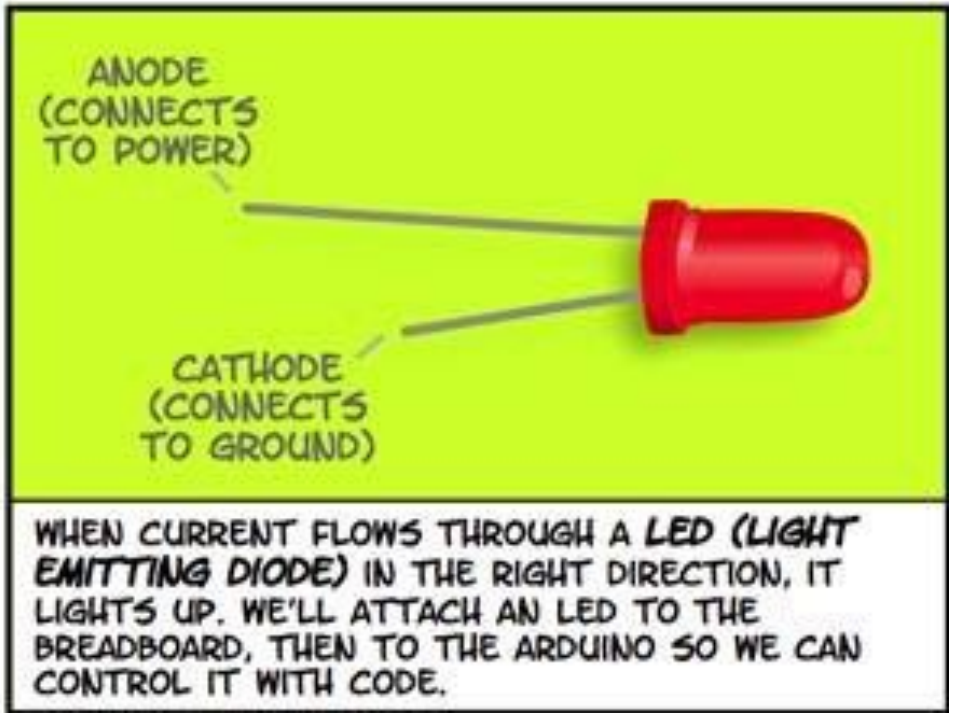
d serial communication and processing

e (optional) I2C communication

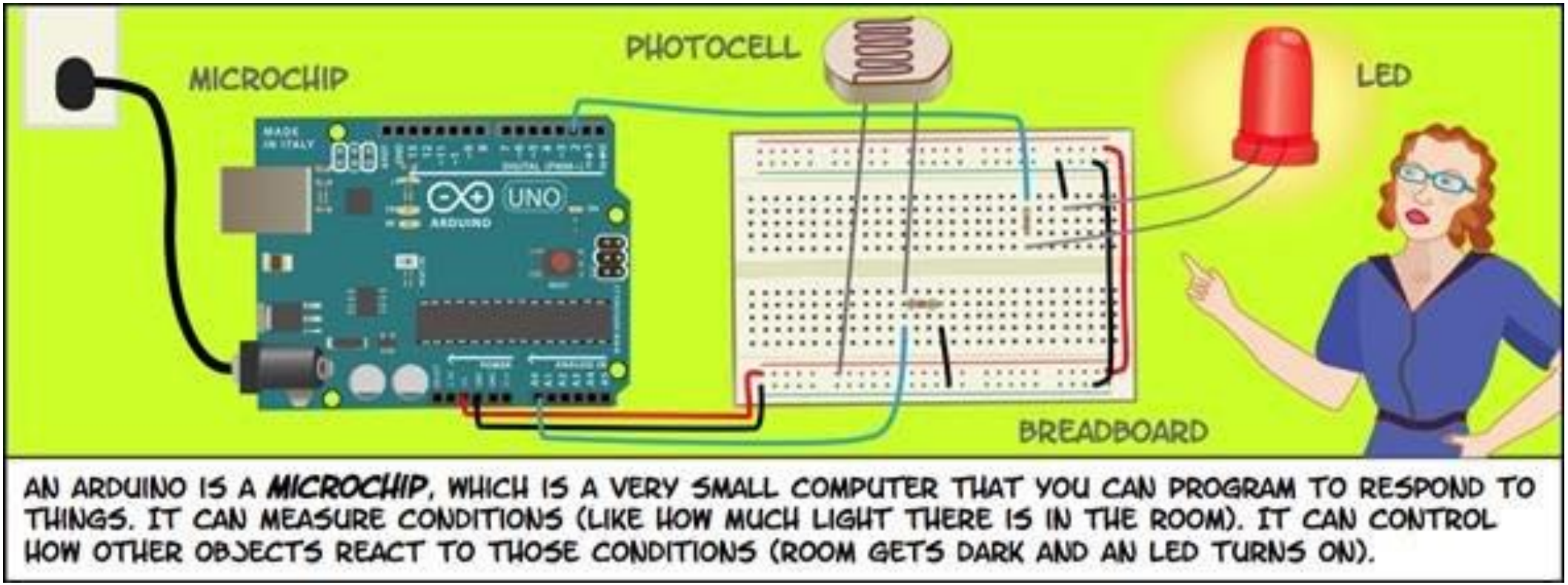






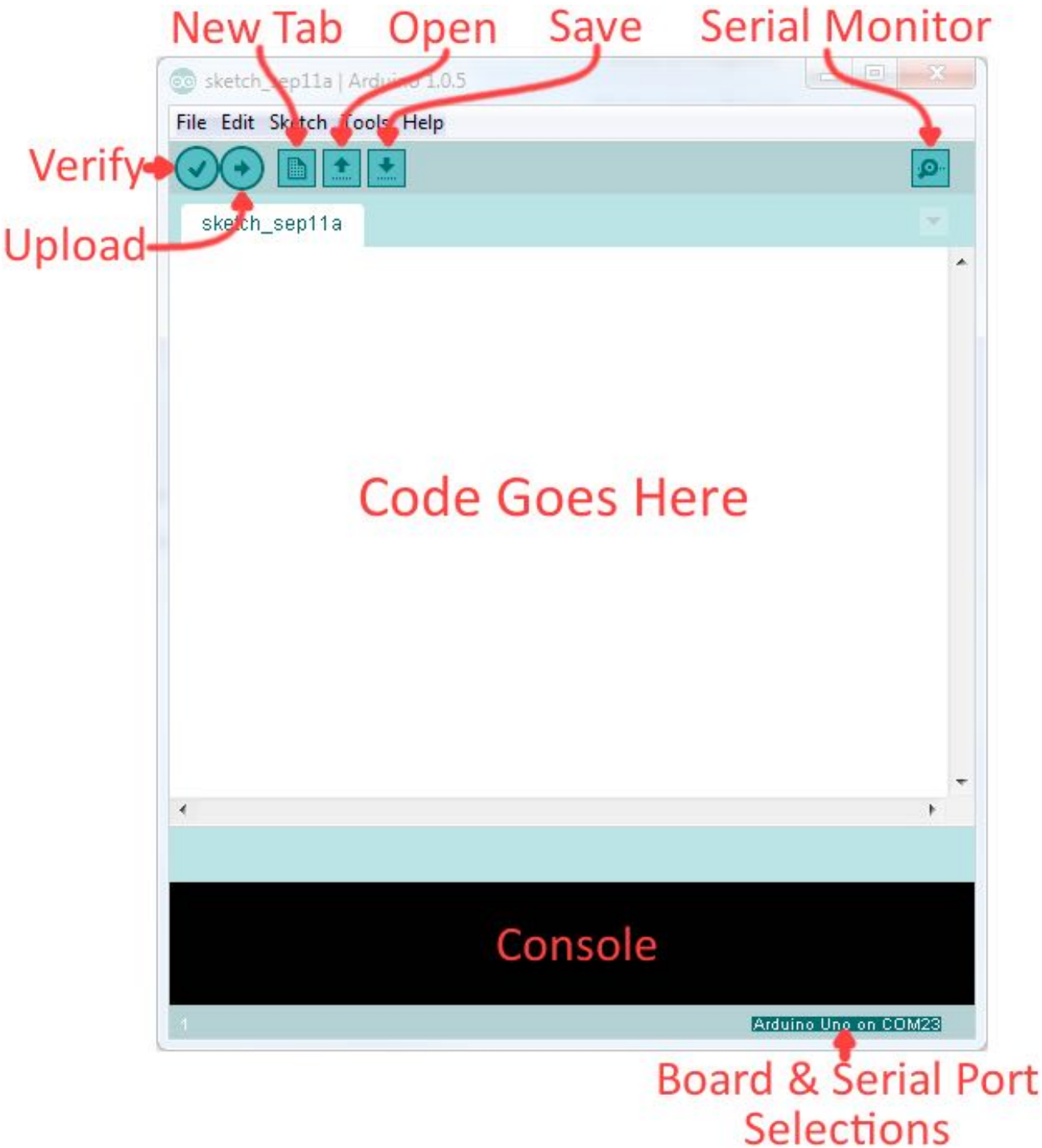


source: http://mods-n-hacks.wonderhowto.com

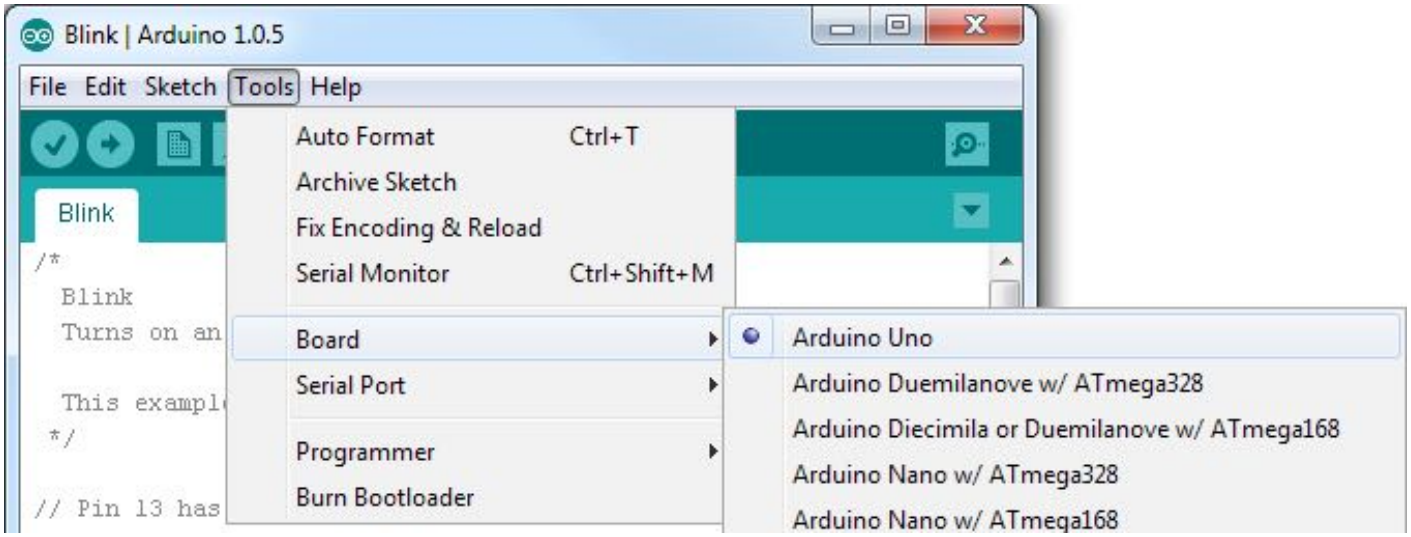


AN ARDUINO IS A ***MICROCHIP***, WHICH IS A VERY SMALL COMPUTER THAT YOU CAN PROGRAM TO RESPOND TO THINGS. IT CAN MEASURE CONDITIONS (LIKE HOW MUCH LIGHT THERE IS IN THE ROOM). IT CAN CONTROL HOW OTHER OBJECTS REACT TO THOSE CONDITIONS (ROOM GETS DARK AND AN LED TURNS ON).

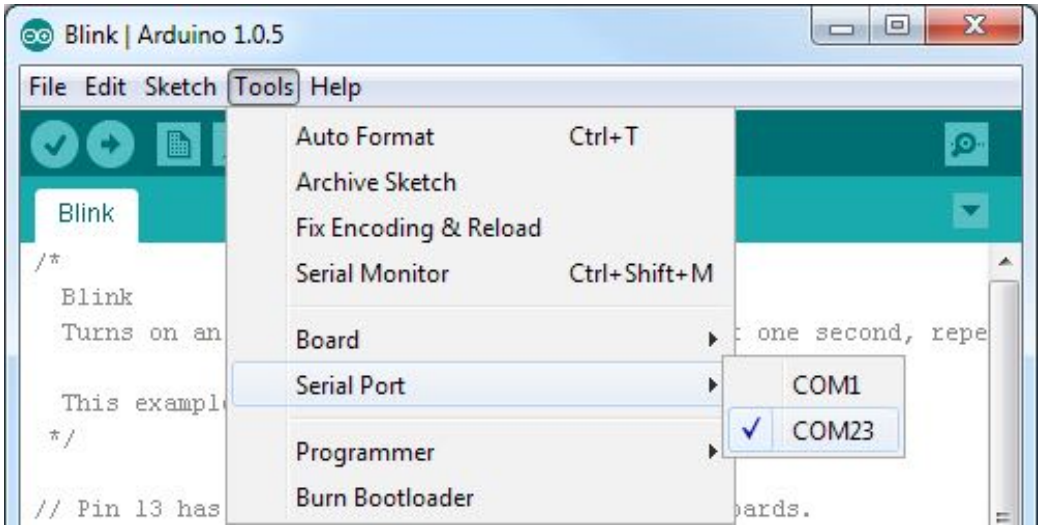
source: http://mods-n-hacks.wonderhowto.com


New Tab
Open
Save
Serial Monitor
Verify
Upload
File Edit Sketch Tools Help
sketch_sep11a
Code Goes Here
Console
Arduino Uno on COM23
Board & Serial Port Selections

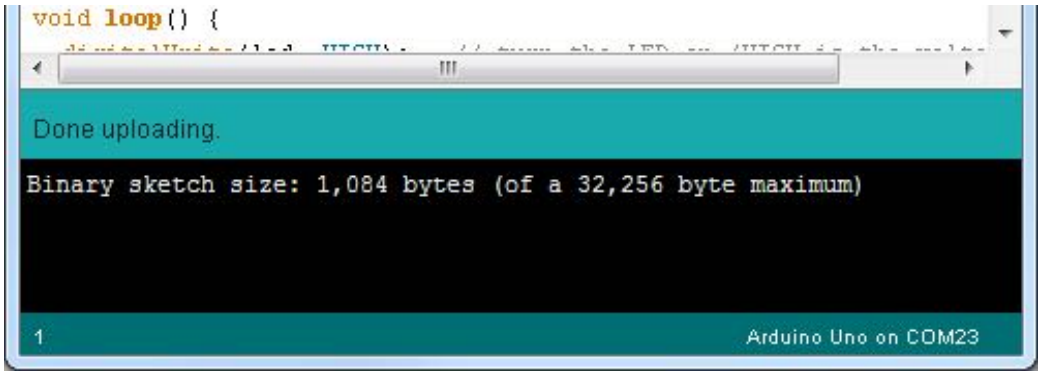
a


Blink | Arduino 1.0.5
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an
This exampl
*/
// Pin 13 has
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168

b

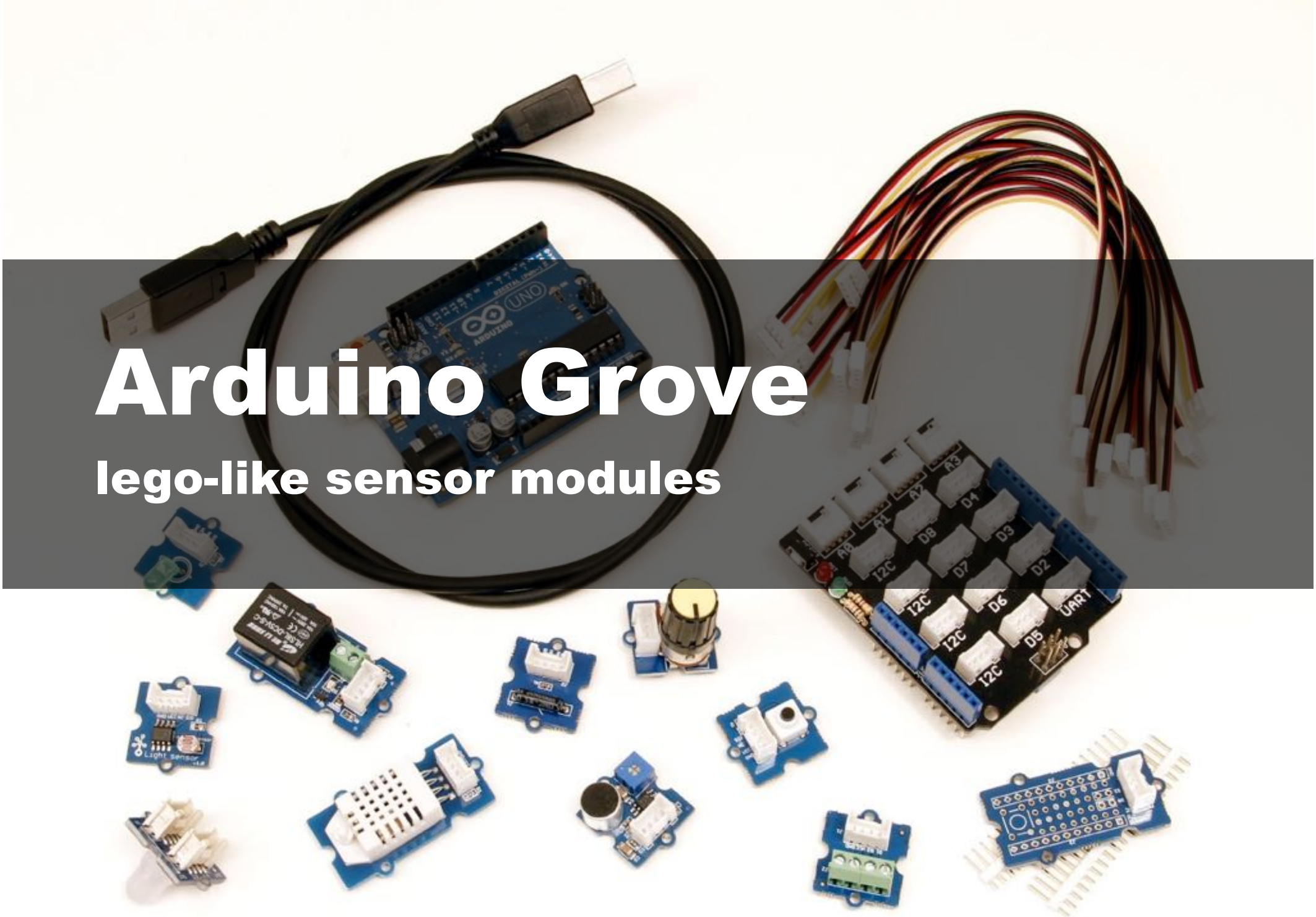

Blink | Arduino 1.0.5
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an
This exampl
*/
// Pin 13 has
one second, repe
ards.
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
COM1
COM23

c


void loop() {
Done uploading.
Binary sketch size: 1,084 bytes (of a 32,256 byte maximum)
1
Arduino Uno on COM23

Arduino Grove

lego-like sensor modules



Grove - 3-Axis Digital Compass

Grove - 3-Axis Digital Accelerometer(±1.5g)

Grove - 3-Axis Digital Gyro

Grove - Collision Sensor

G

Grove - 3-Axis Digital Accelerometer(±16g)

Grove - 6-Axis Accelerometer and Compass V1.0

Grove - Single Axis Analog Gyro

/ireless Communication

ommunicating without wires is a cool feature that can spice up your project. Modules in this category arm your microcontroller with wireless commu

variable resistance

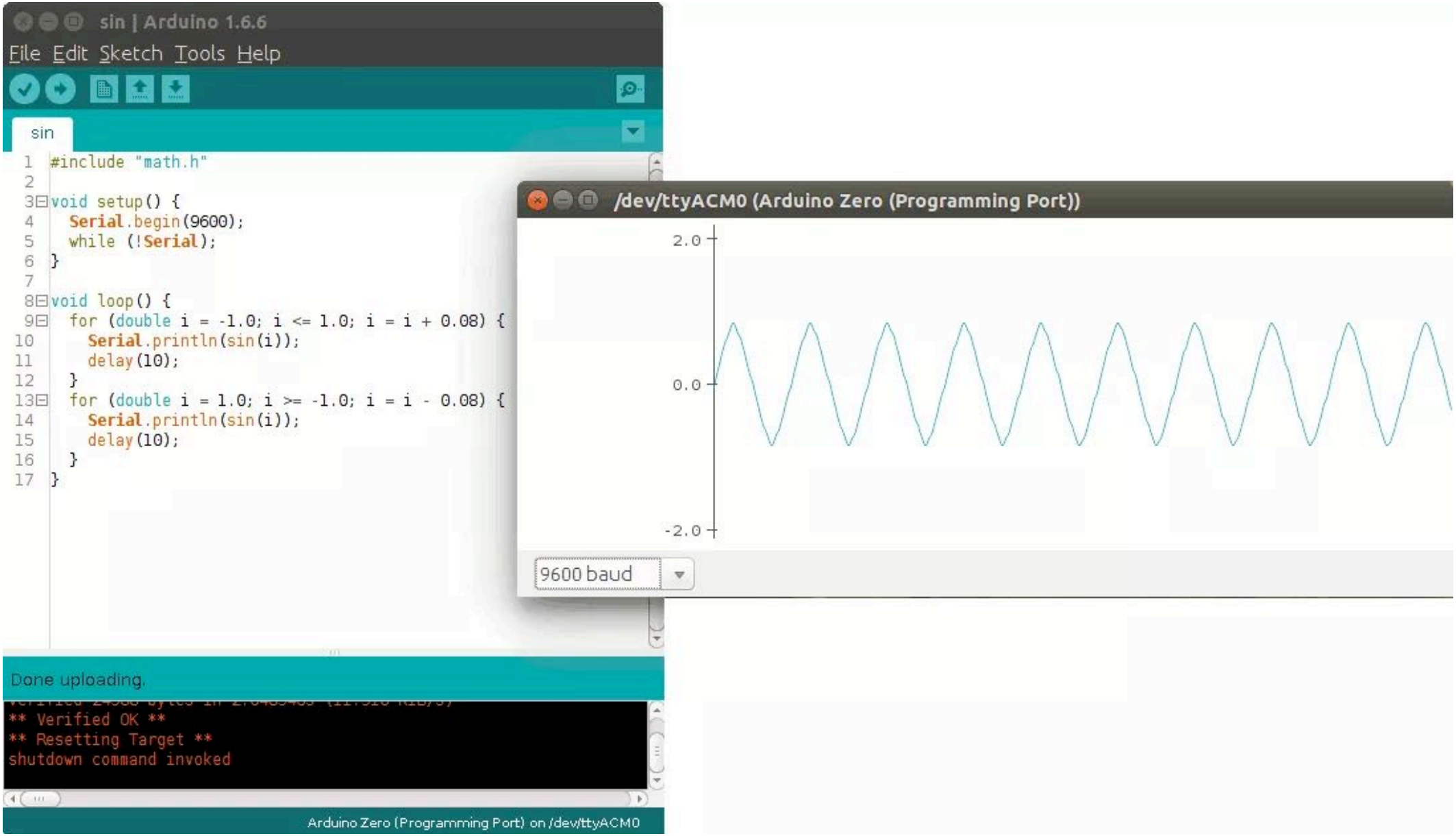
_01_variable_resistance | Arduino 1.0.2

_01_variable_resistance

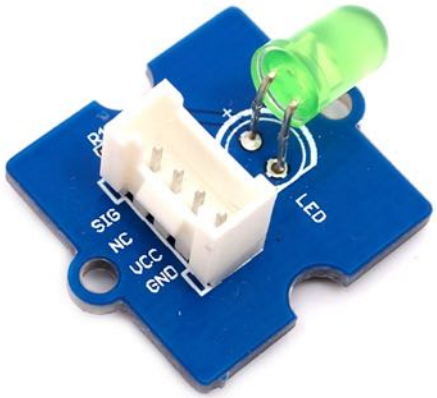
```
int pin0 = 0;
int value = 0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  value = analogRead(pin0);
  delay(30);
  Serial.println(value);
}
```

1 Arduino Uno on /dev/tty.usbmodem1411

Arduino Plotter



Blinking LED



_02_led_blink | Arduino 1.0.2

_02_led_blink

```
#define LED 2 //connect LED to digital pin2
void setup() {
  // initialize the digital pin2 as an output.
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(LED, HIGH);   // set the LED on
  delay(500);               // for 500ms
  digitalWrite(LED, LOW);   // set the LED off
  delay(500);
  Serial.println("hahaha");
}
```

1 Arduino Uno on /dev/tty.usbmodem1411

Touch Sensor with LED

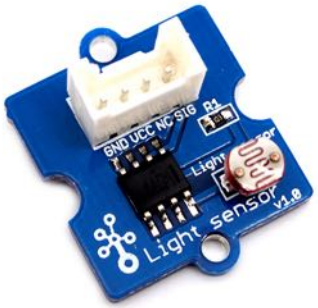
_04_touch_led | Arduino 1.0.2

_04_touch_led

```
const int TouchPin=1;
const int ledPin=2;
void setup() {
pinMode(TouchPin, INPUT);
pinMode(ledPin,OUTPUT);
}

void loop() {
  int sensorValue = digitalRead(TouchPin);
  if(sensorValue==1) {
    digitalWrite(ledPin,HIGH);
  } else {
    digitalWrite(ledPin,LOW);
  }
}
```

1 Arduino Uno on /dev/tty.usbmodem1411

Light Sensor with LED



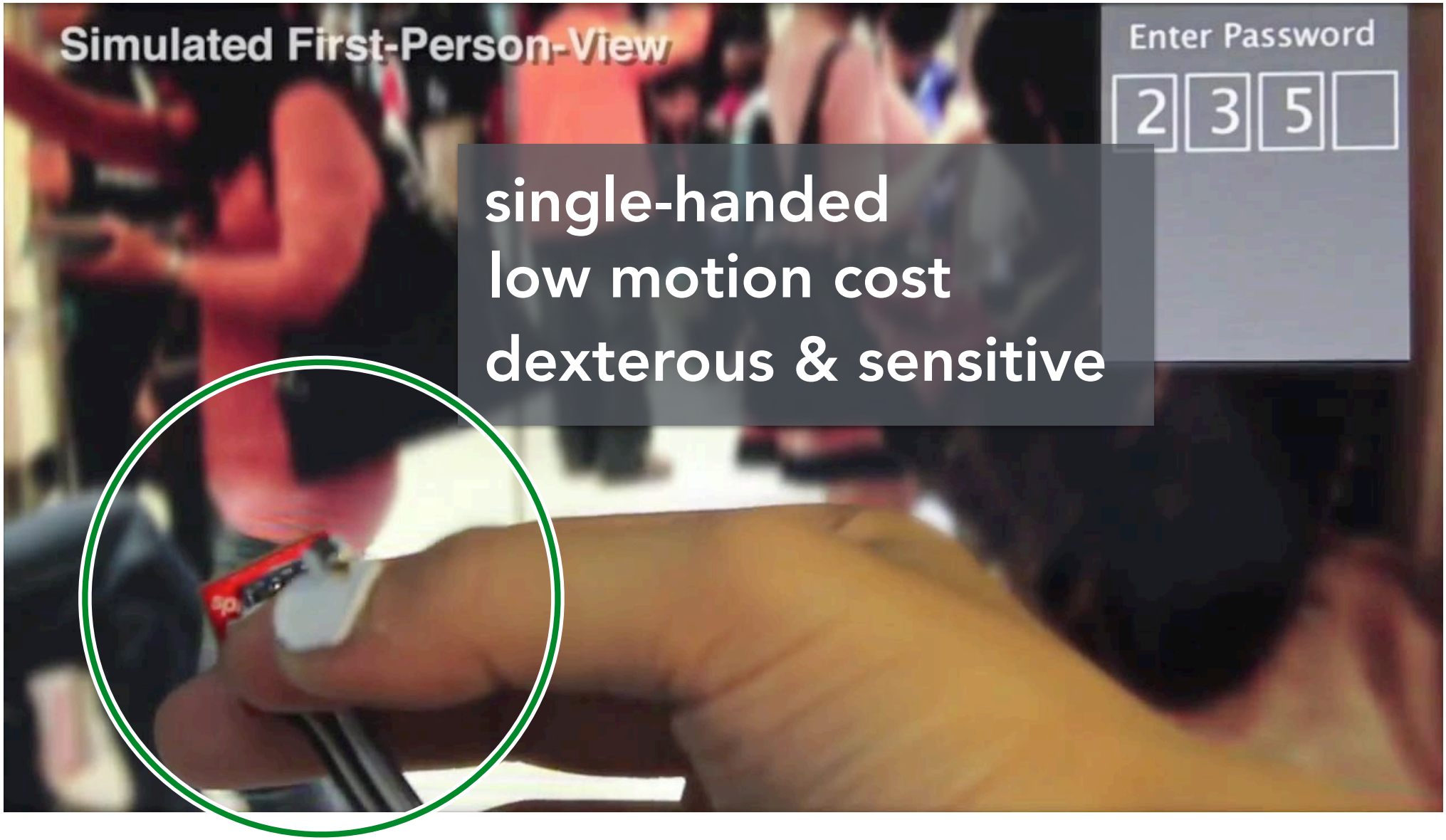
_05_light_sensor_led | Arduino 1.0.2

_05_light_sensor_led

```
#include <math.h>
const int ledPin=2;                  //Connect the LED Grove module to Pin12, Dig
const int thresholdvalue=10;          //The treshold for which the LED should tur
float Rsensor; //Resistance of sensor in K
void setup() {
  Serial.begin(9600);                 //Start the Serial connection
  pinMode(ledPin,OUTPUT);             //Set the LED on Digital 12 as an OUTPUT
}
void loop() {
  int sensorValue = analogRead(1);
  Rsensor=(float)(1023-sensorValue)*10/sensorValue;
  if(Rsensor>thresholdvalue)
  {
    digitalWrite(ledPin,HIGH);
  }
  else
  {
  digitalWrite(ledPin,LOW);
  }
  Serial.println("the analog read data is ");
  Serial.println(sensorValue);
  Serial.println("the sensor resistance is ");
  Serial.println(Rsensor,DEC);//show the ligth intensity on the serial monitor;
  delay(500);

}
```

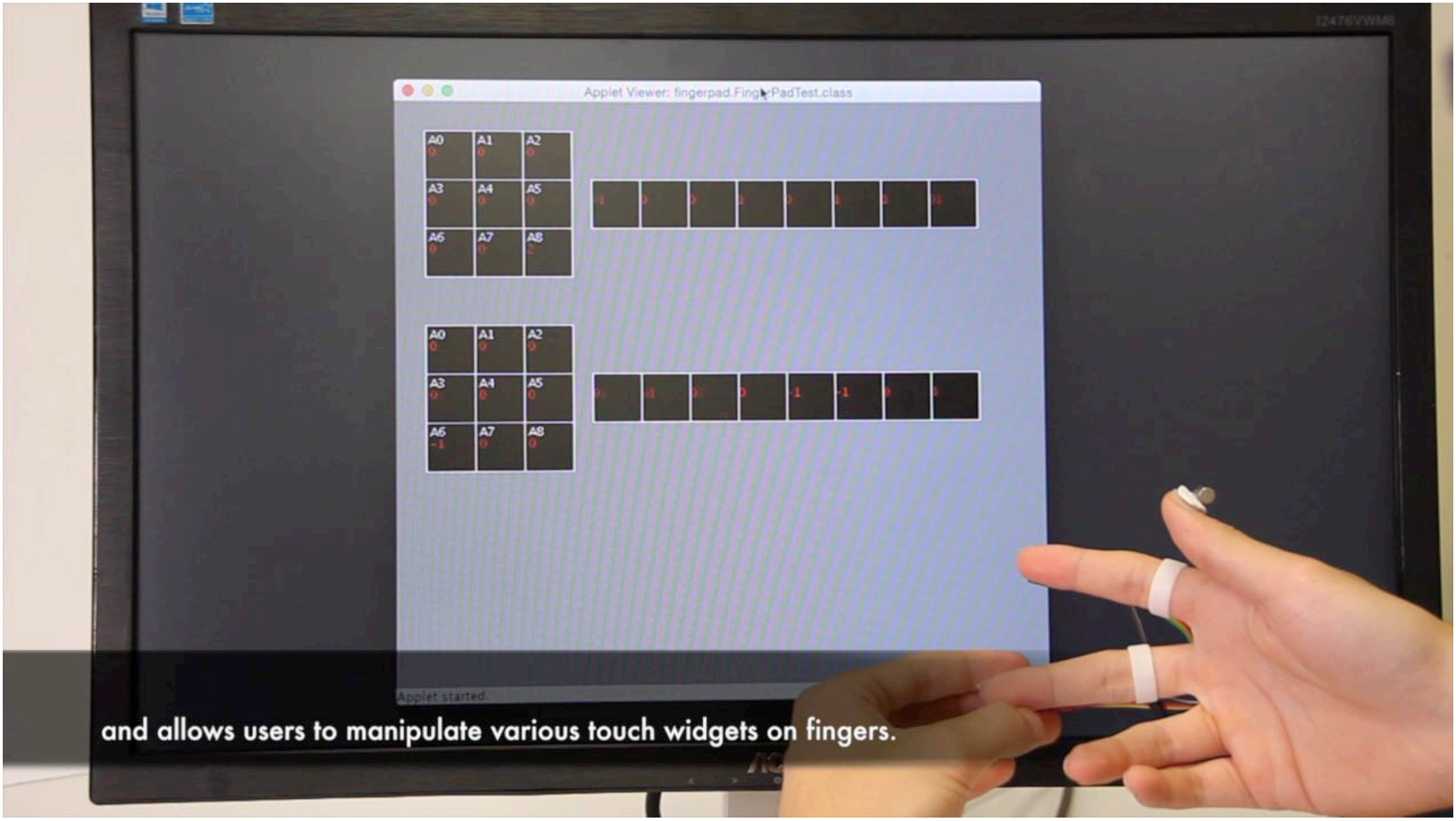
1 Arduino Uno on /dev/tty.usbmodem1411

FingerPad

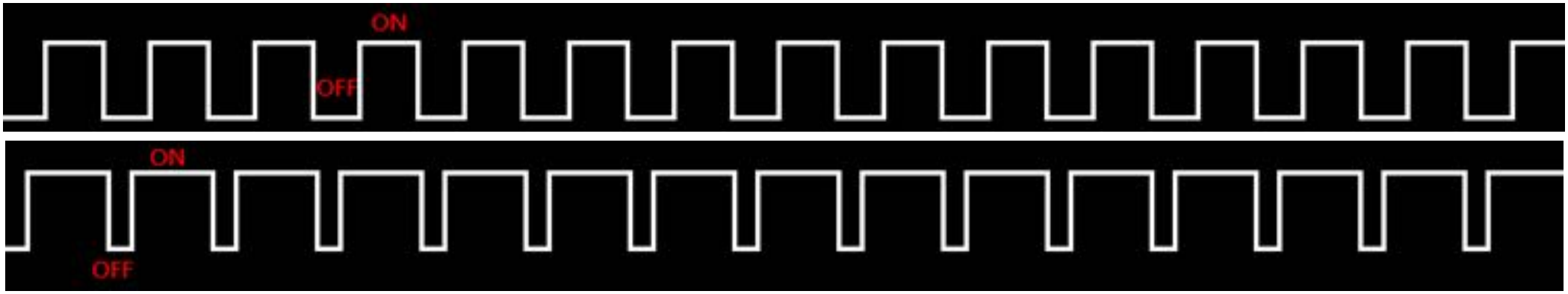


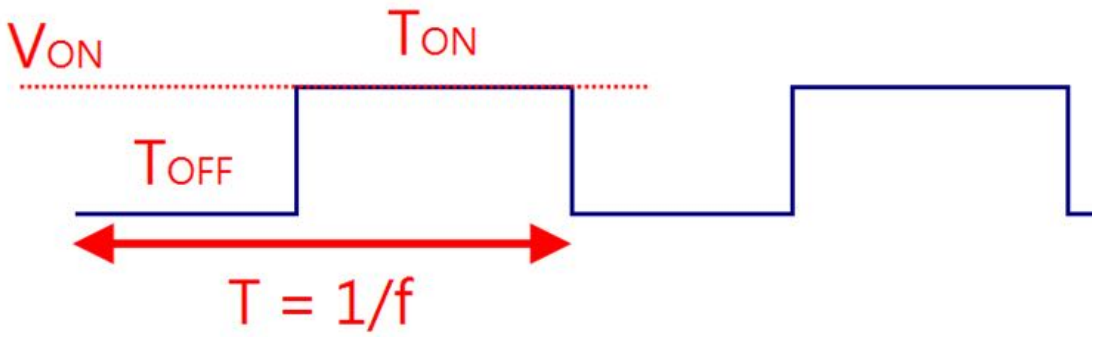
single-handed
low motion cost
dexterous & sensitive

DigitSpace


Applet Viewer: fingerpad.FingerPadTest.class
A0
A1
A2
A3
A4
A5
A6
A7
A8
Applet started.
and allows users to manipulate various touch widgets on fingers.

Analog Output of Arduino: Pulse Width Modulation (PWM)

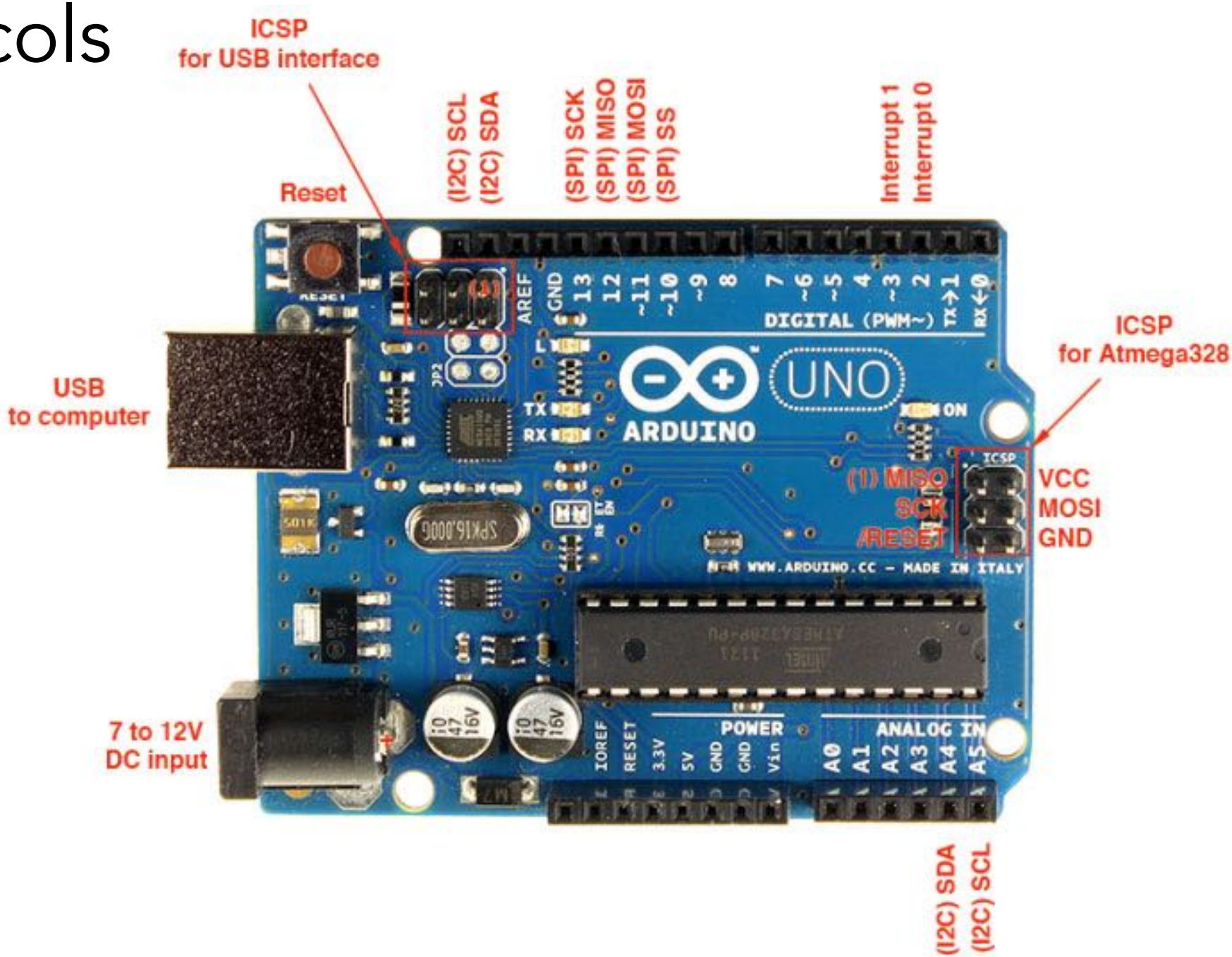




Duty Cycle = Toff / Ton (%)

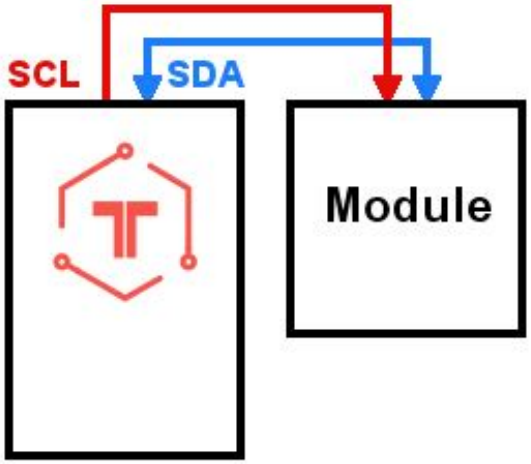
Vsim = Von x Duty Cycle

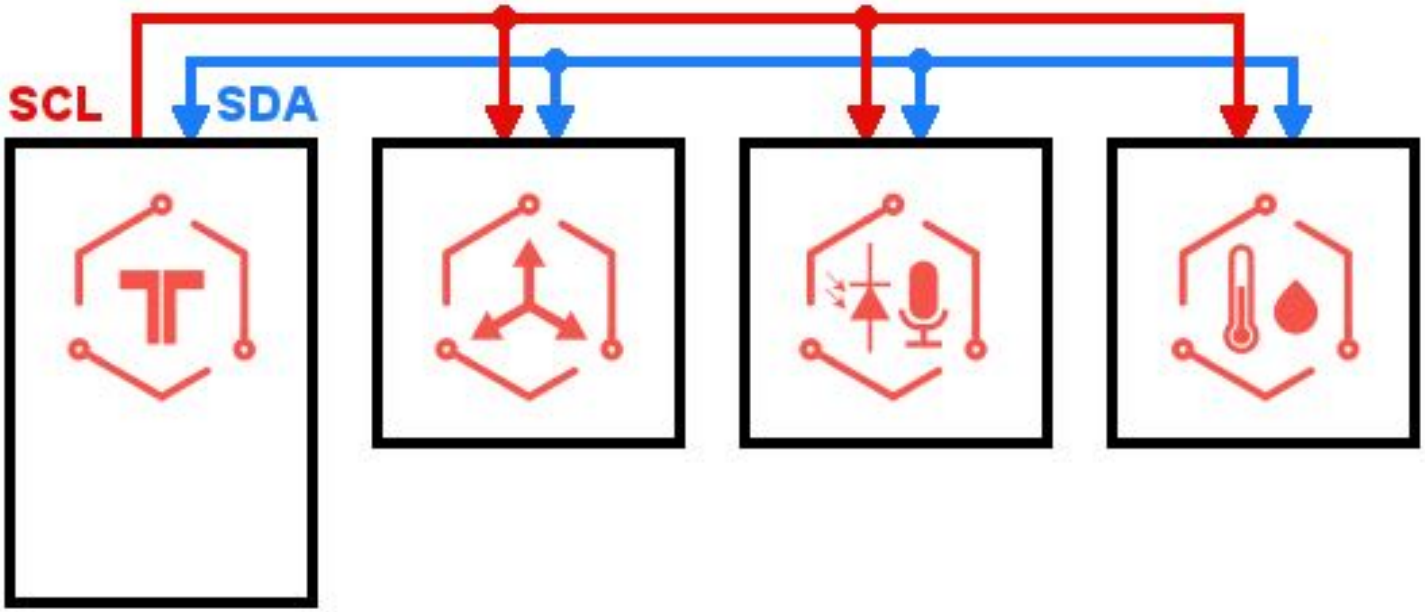
Other Protocols



I²C

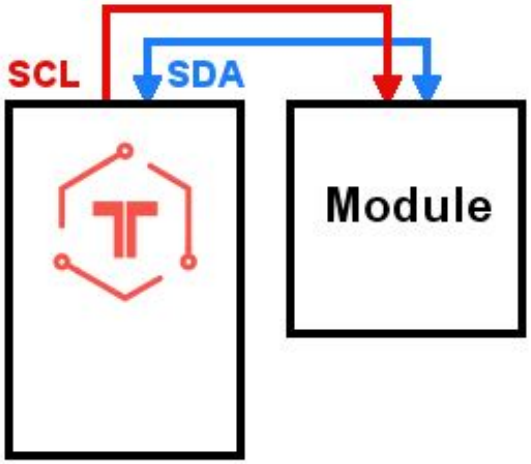
Inter-Integrated Circuits

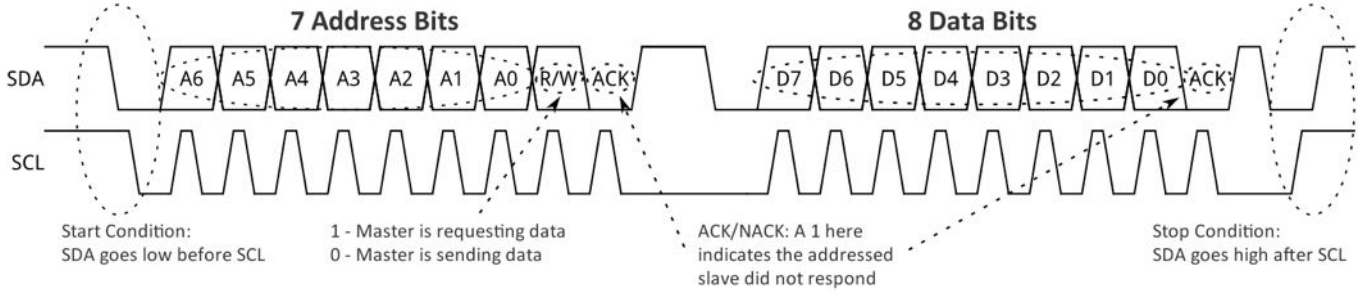


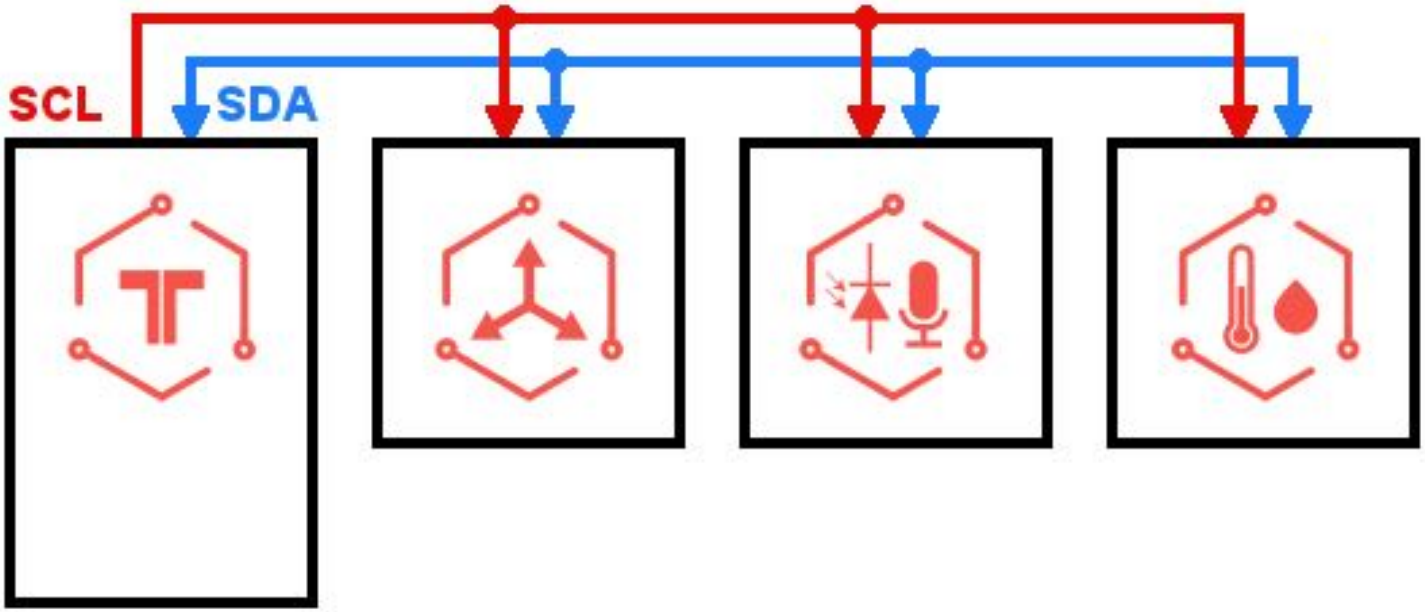


I²C

Inter-Integrated Circuits



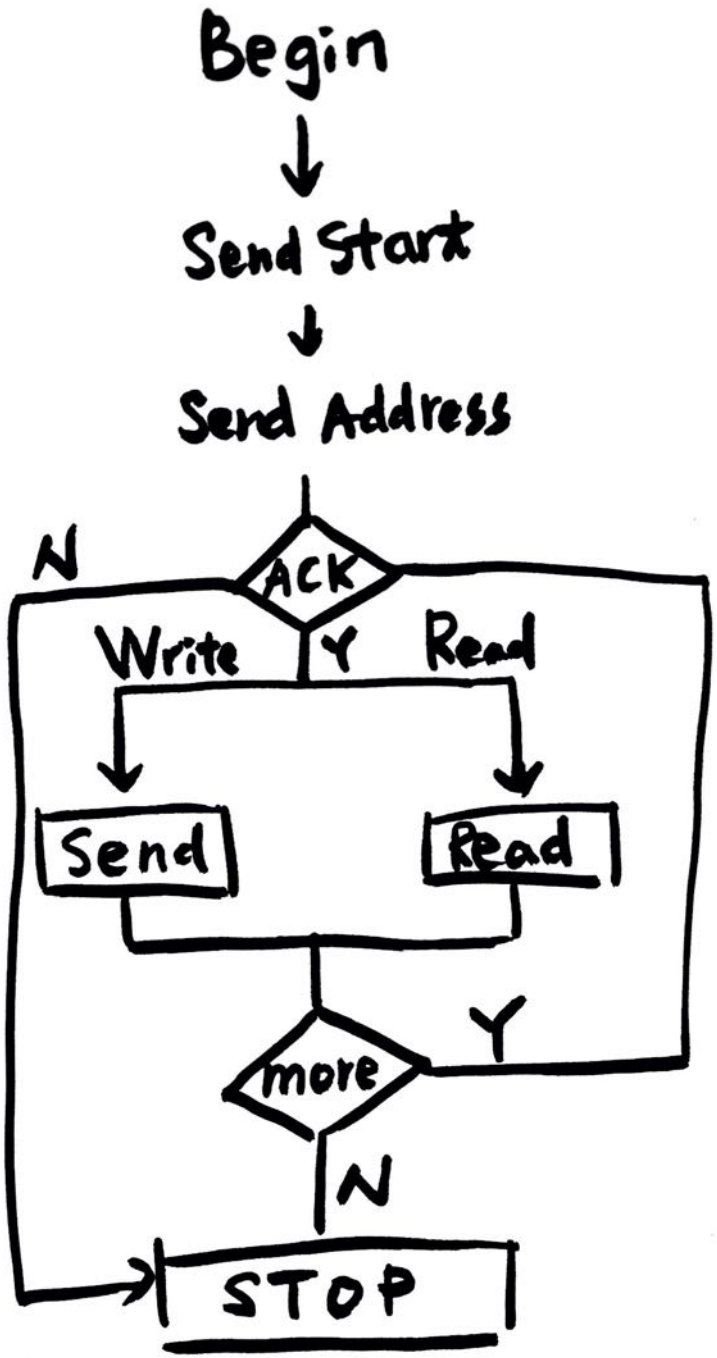




Communication Process

I²C

Inter-Integrated Circuits



I²C

Inter-Integrated Circuits

Read Byte Format

S	Address	WR	ACK	Command	ACK	S	Address	RD	ACK	Data	NACK	P
	7 Bits			8 Bits			7 Bits			8 Bits		

Slave Address

Command Byte: selects which register you are reading from.

Slave Address: repeated due to change in data-flow direction.

Data Byte: reads from the register set by the command byte.

```
byte readVCNLByte(byte address){
  // readByte(address) reads a single byte of data from address
  Wire.beginTransmission(VCNL4000_ADDRESS);
  Wire.write(address);
  Wire.endTransmission();
  Wire.requestFrom(VCNL4000_ADDRESS, 1);
  while(!Wire.available());
  byte data = Wire.read();

  return data;
}
```

I²C

Inter-Integrated Circuits

Write Byte Format

S	Address	WR	ACK	Command	ACK	Data	ACK	P
	7 Bits			8 Bits		8 Bits		

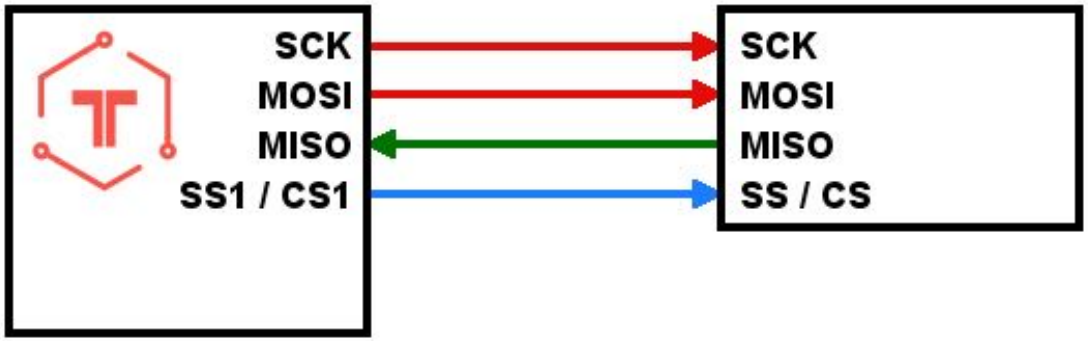
Slave Address

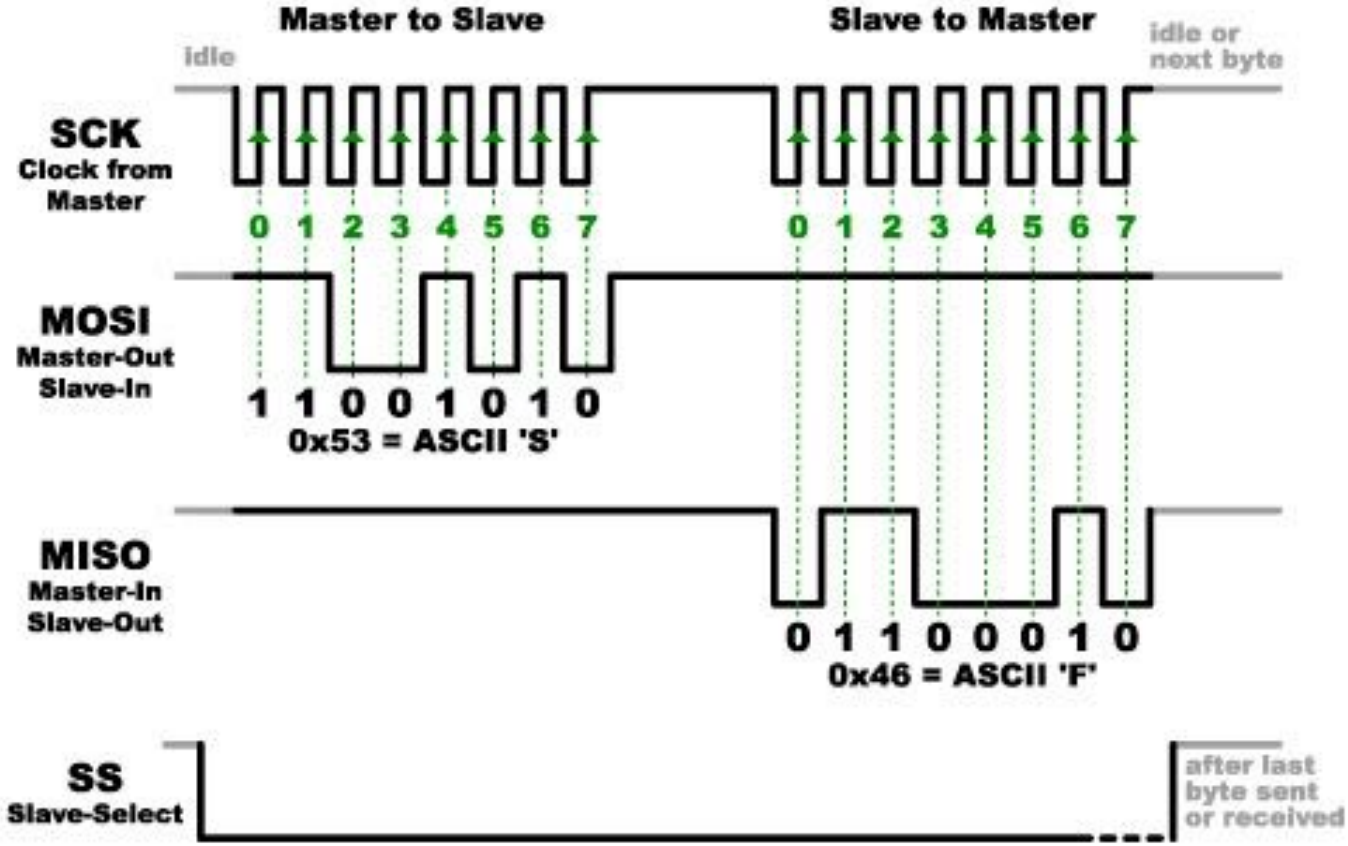
Command Byte: selects which register you are writing to.

Data Byte: data goes into the register set by the command byte.

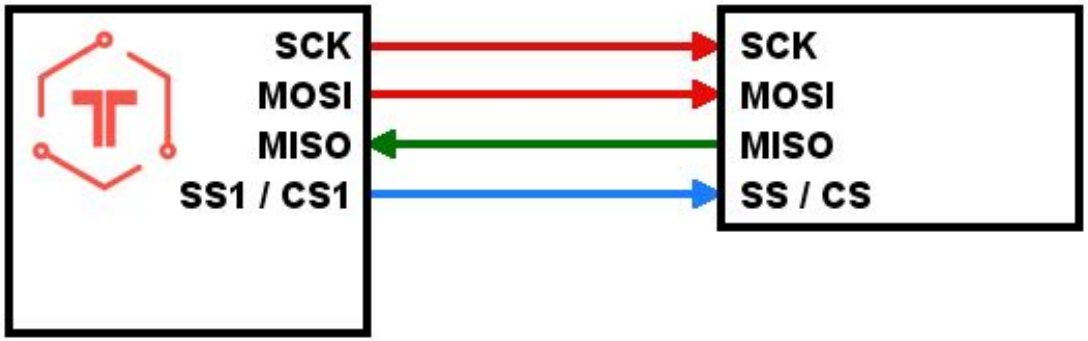
```
void writeVCNLByte(byte address, byte data){
  // writeVCNLByte(address, data) writes a single byte of data to address
  Wire.beginTransmission(VCNL4000_ADDRESS);
  Wire.write(address);
  Wire.write(data);
  Wire.endTransmission();
}
```

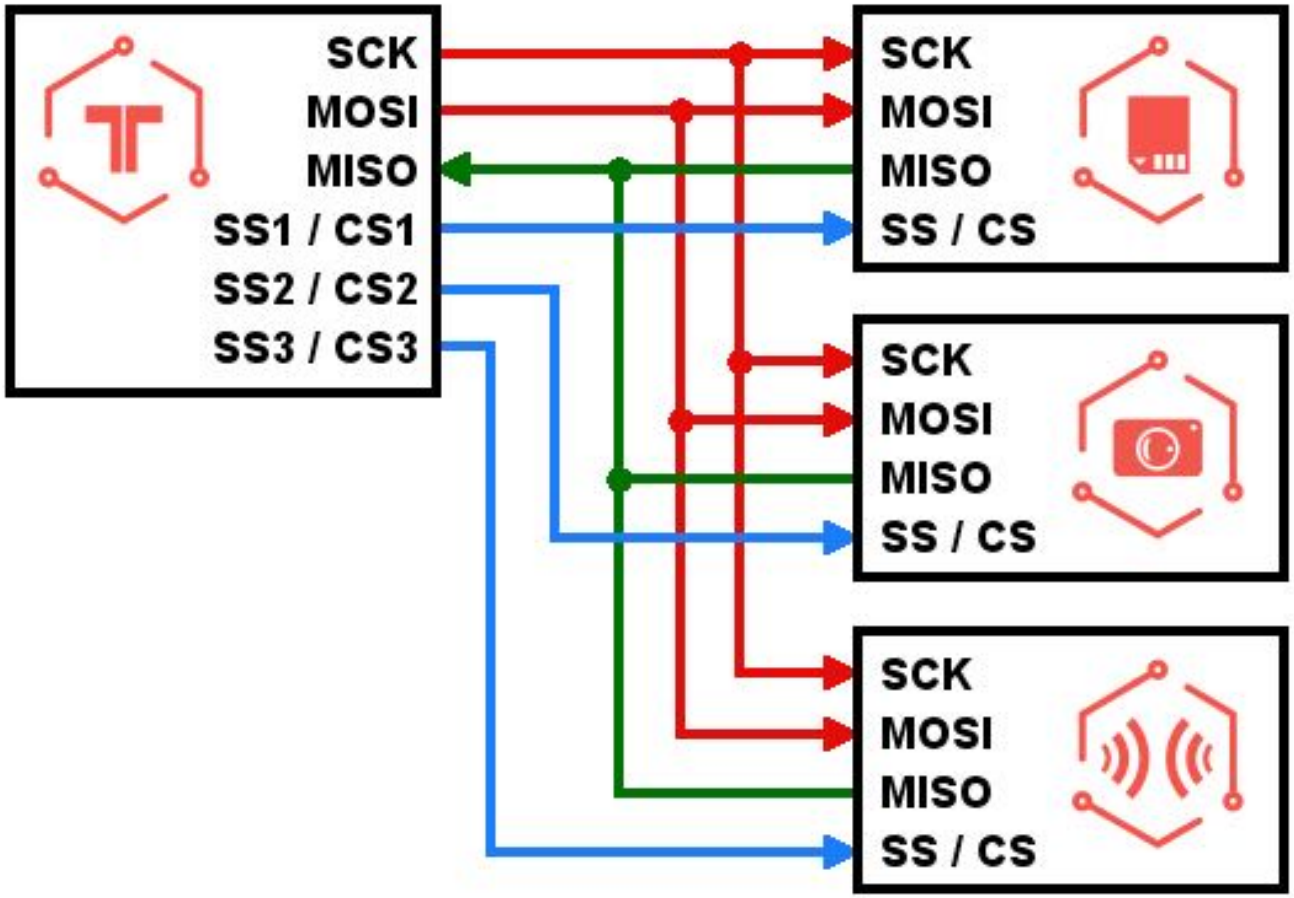
SPI


SCK
MOSI
MISO
SS1 / CS1
SCK
MOSI
MISO
SS / CS


Master to Slave
Slave to Master
idle
idle or next byte
SCK
Clock from Master
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
MOSI
Master-Out Slave-In
1 1 0 0 1 0 1 0
0x53 = ASCII 'S'
MISO
Master-In Slave-Out
0 1 1 0 0 0 1 0
0x46 = ASCII 'F'
SS
Slave-Select
after last byte sent or received

SPI


SCK
MOSI
MISO
SS1 / CS1
SCK
MOSI
MISO
SS / CS


SCK
MOSI
MISO
SS1 / CS1
SS2 / CS2
SS3 / CS3
SCK
MOSI
MISO
SS / CS
SCK
MOSI
MISO
SS / CS
SCK
MOSI
MISO
SS / CS