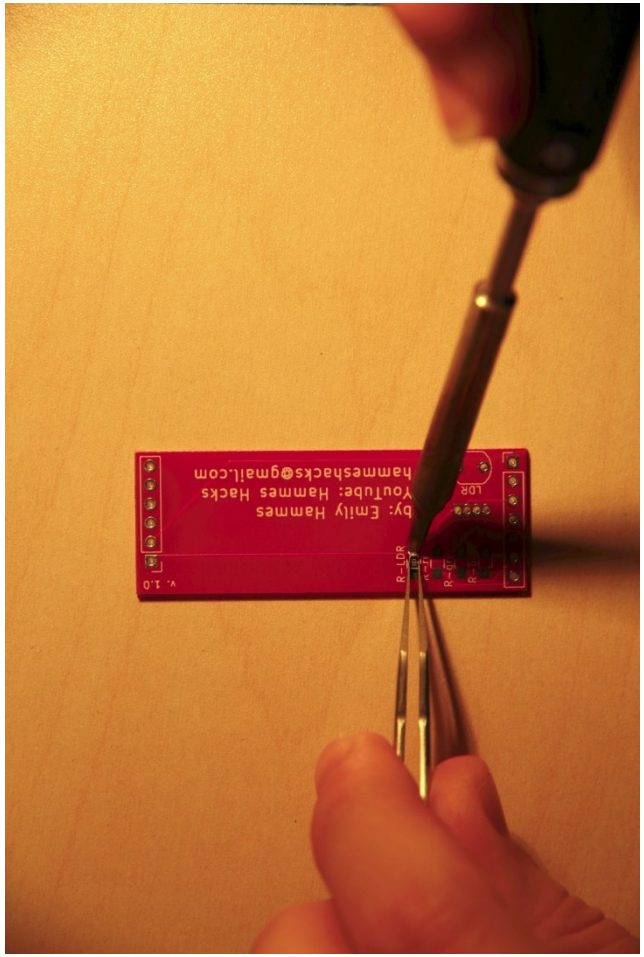
Hammes Hacks
Intro to Arduino

Emily Hammes
hammeshacks@gmail.com
Hammeshacks.com
Youtube: hammeshacks

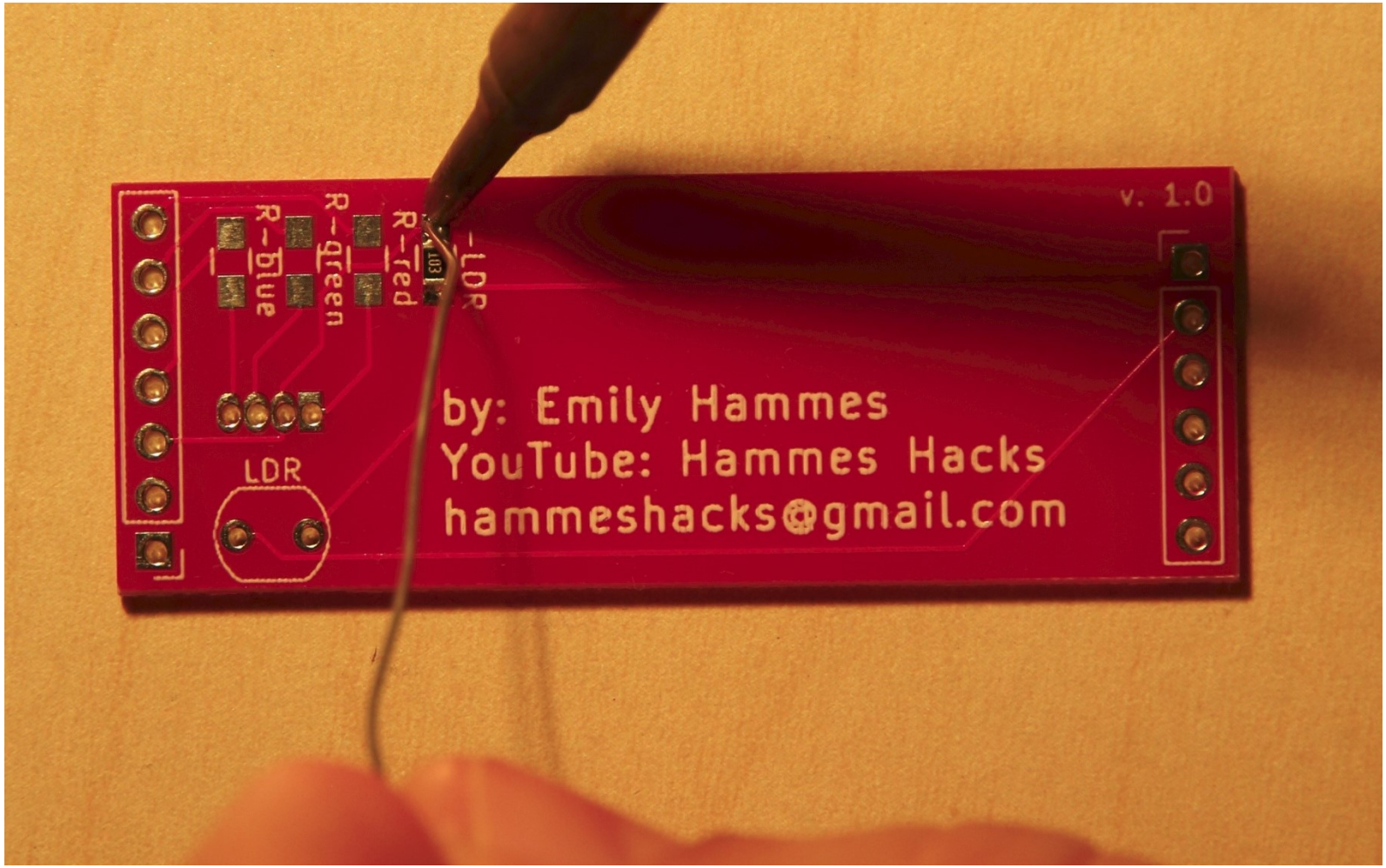
SOLDERING THE SHIELD

Add a Small Amount of Solder to One of the R-LDR pads

Place the Resistor onto the Pad (103 or 10K ohm)

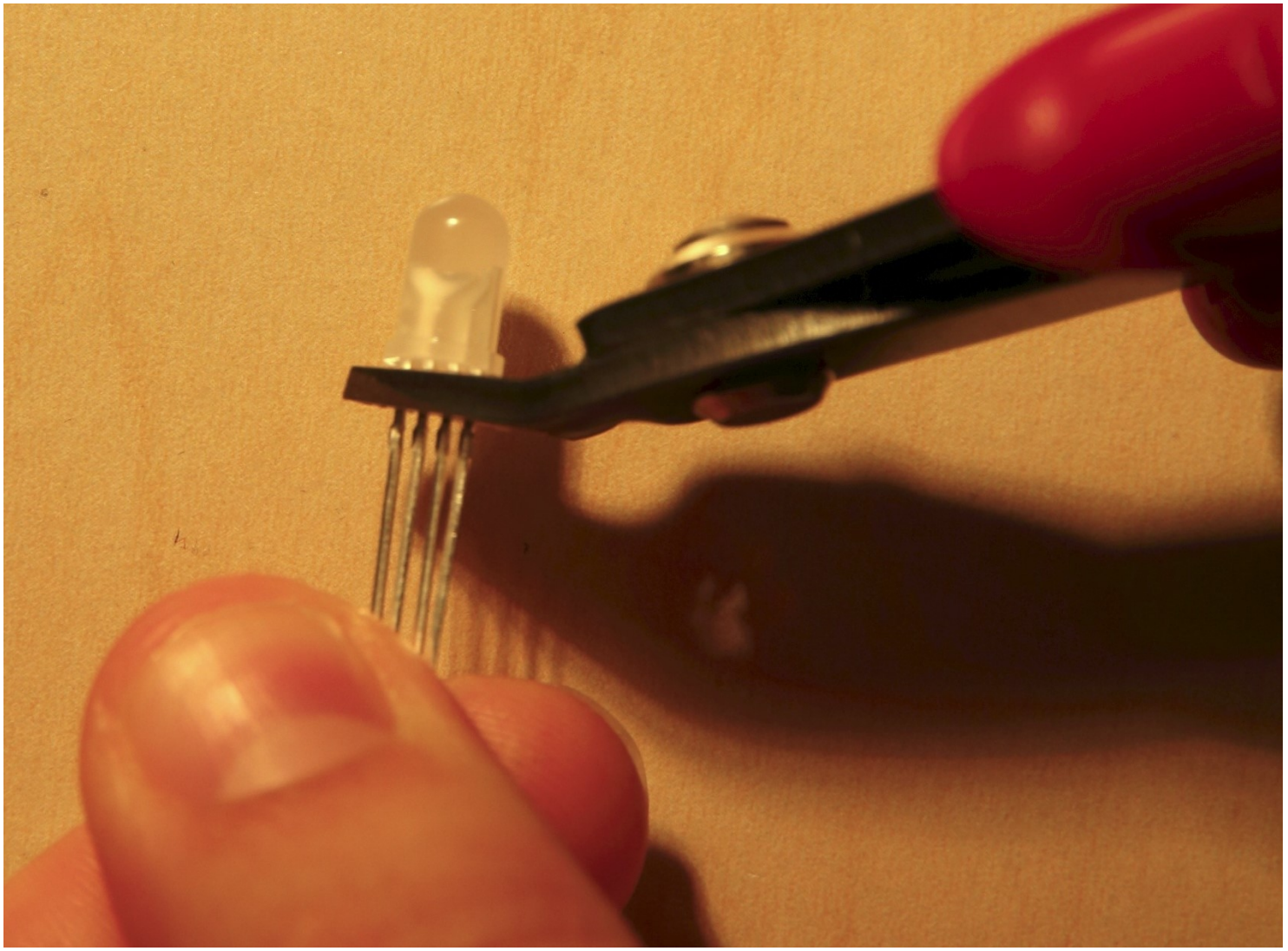


Solder the Other Side of the Resistor



Do the Same for the Red, Green and Blue Resistors (221 or 220)

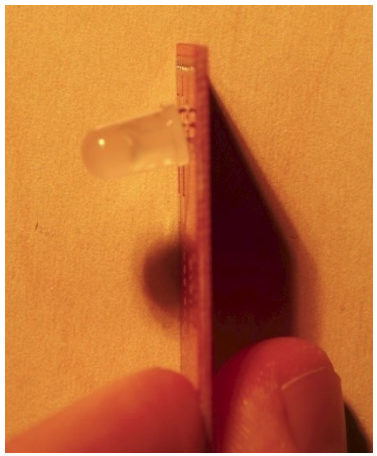
Trim the Leads on the RGB LED Leaving 2 mm

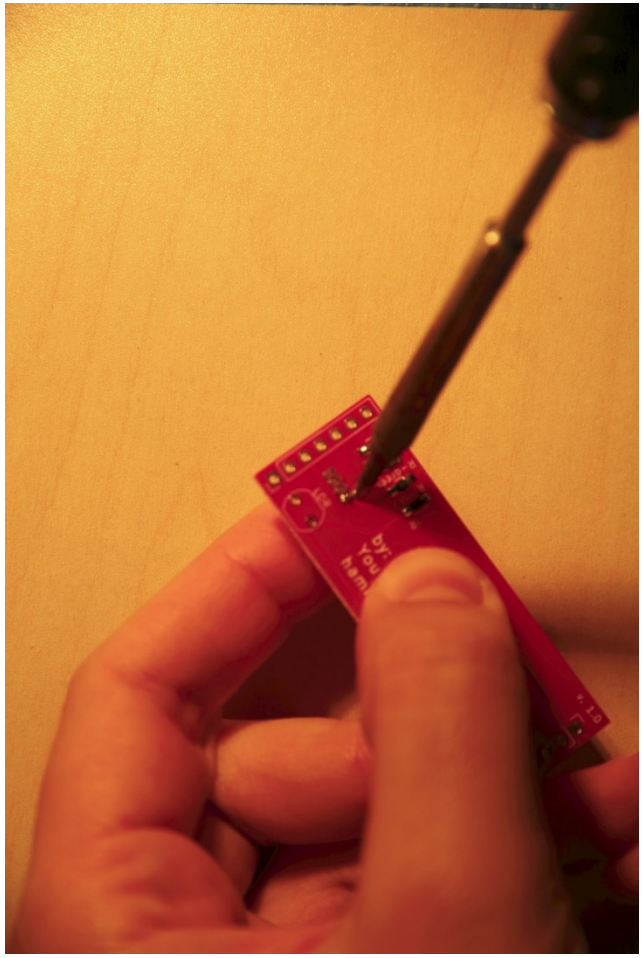


Insert the LED with the Flat Side Matching the Silk Screen

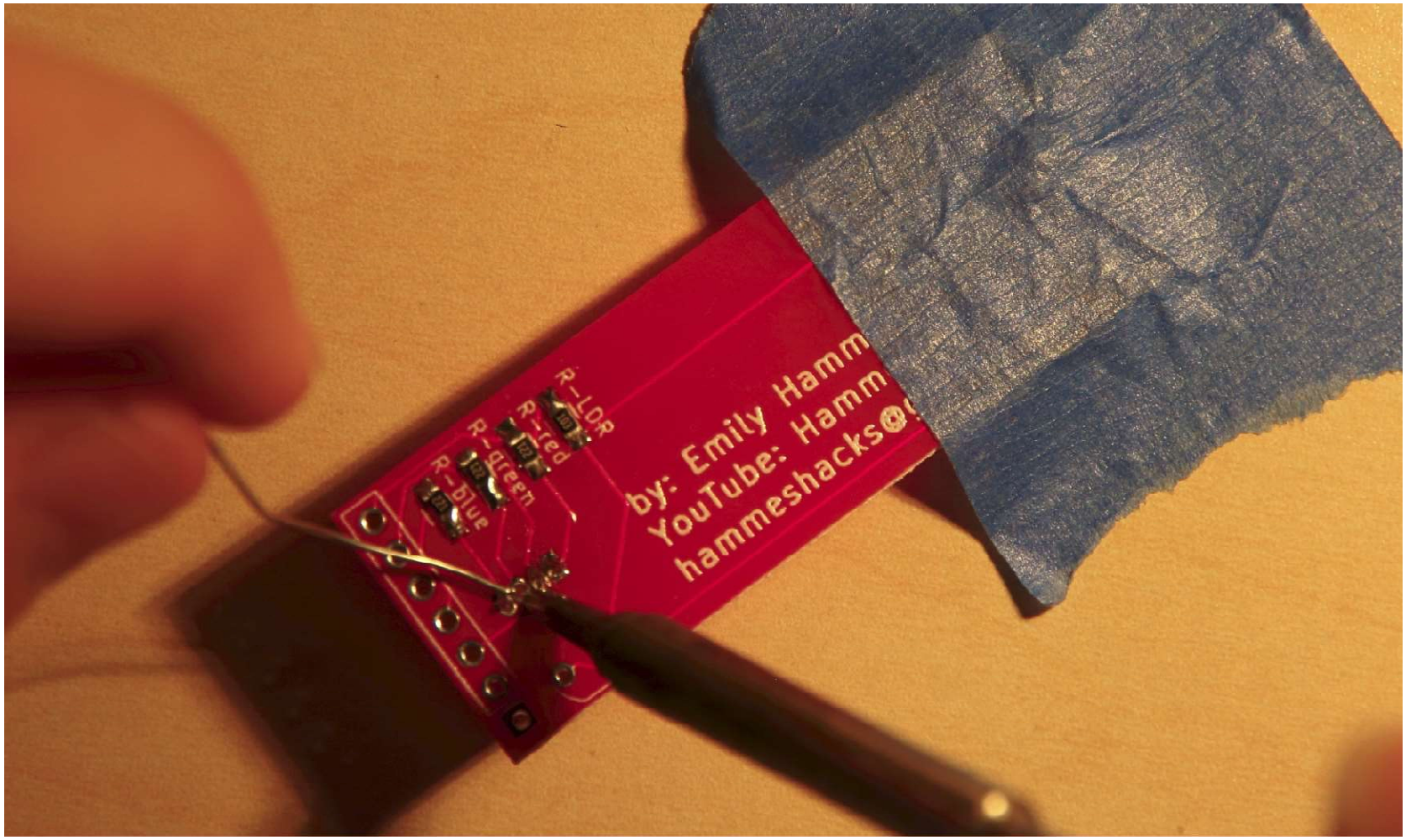
Solder the Square Pad

Realign if Necessary

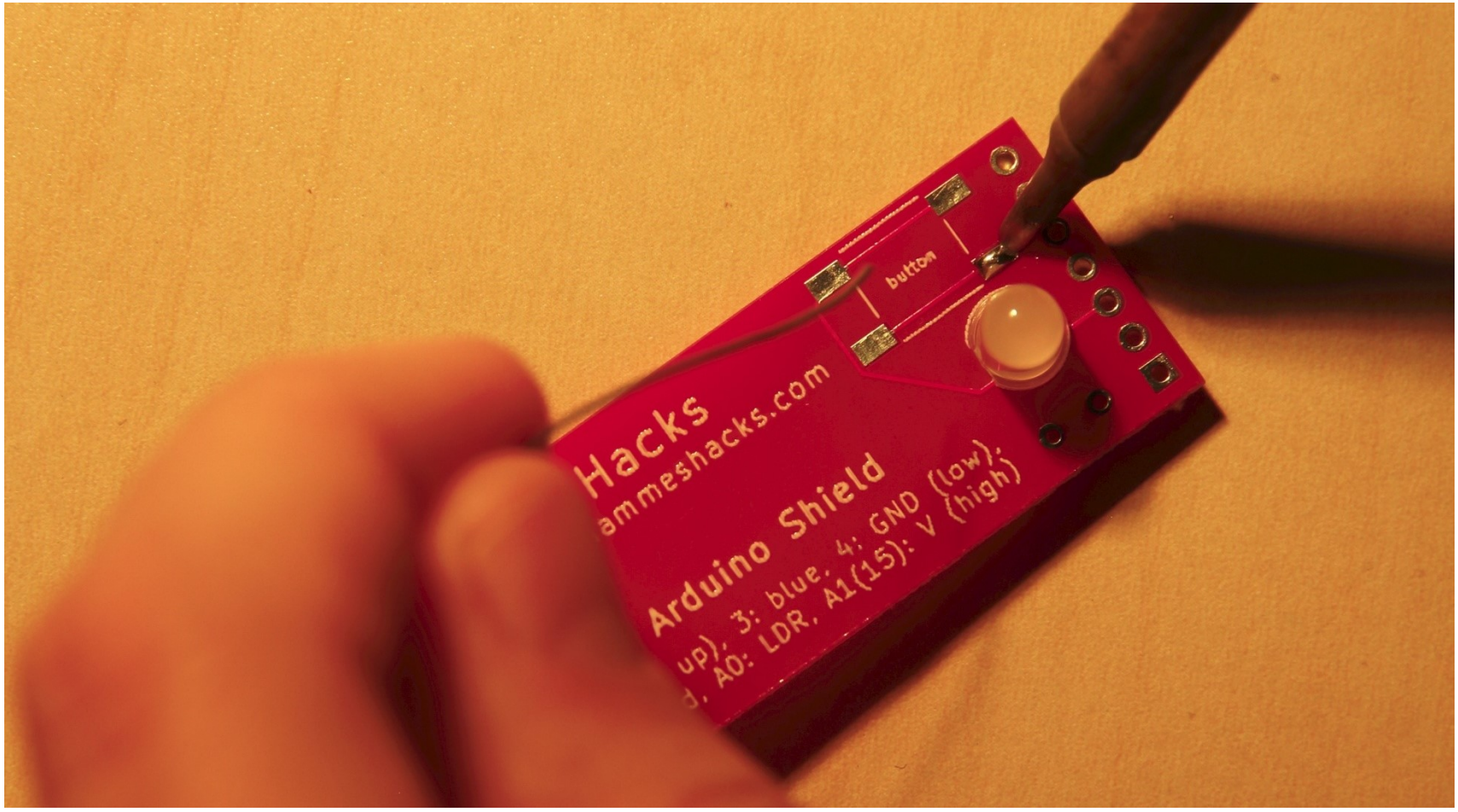




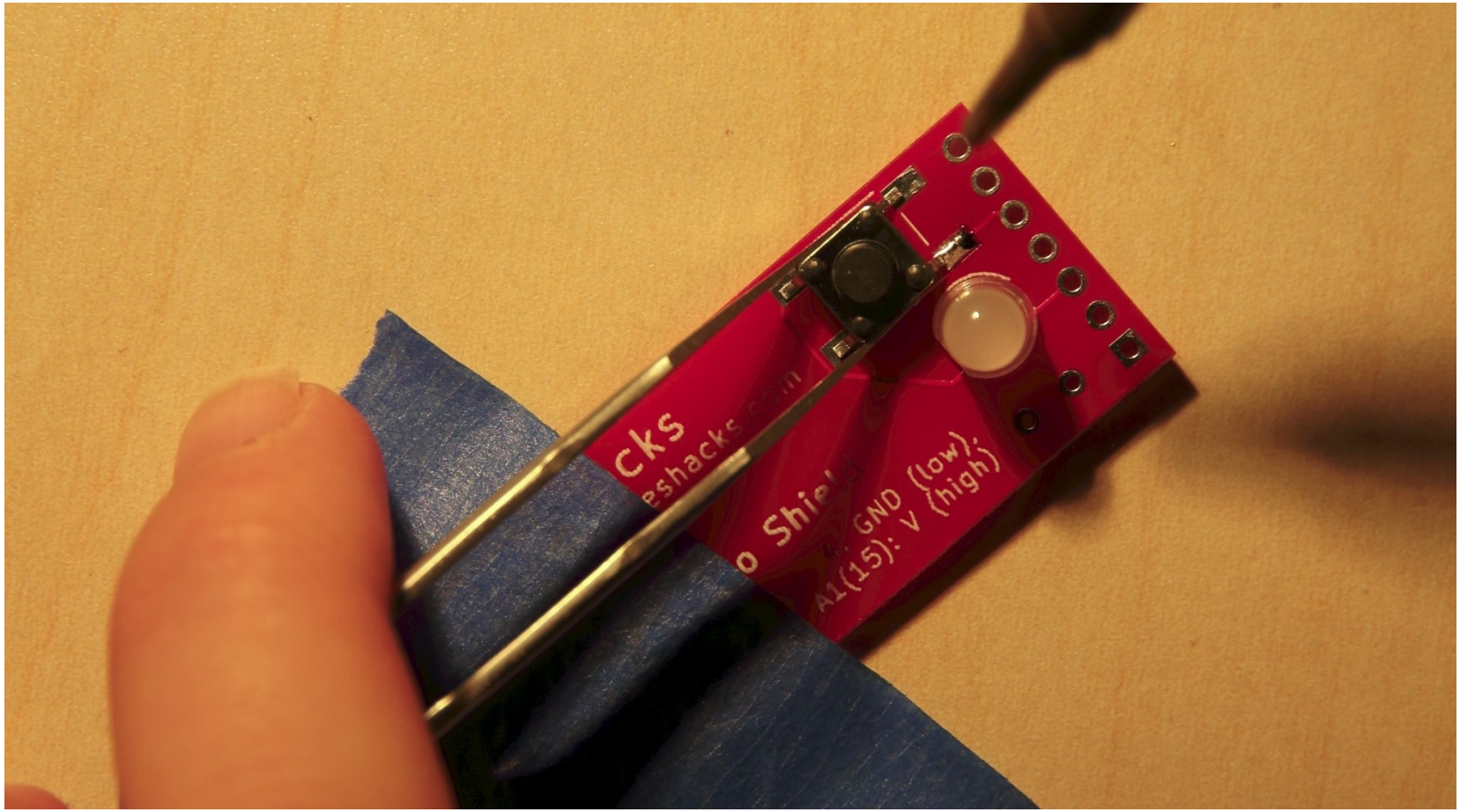
Solder the Other Pins



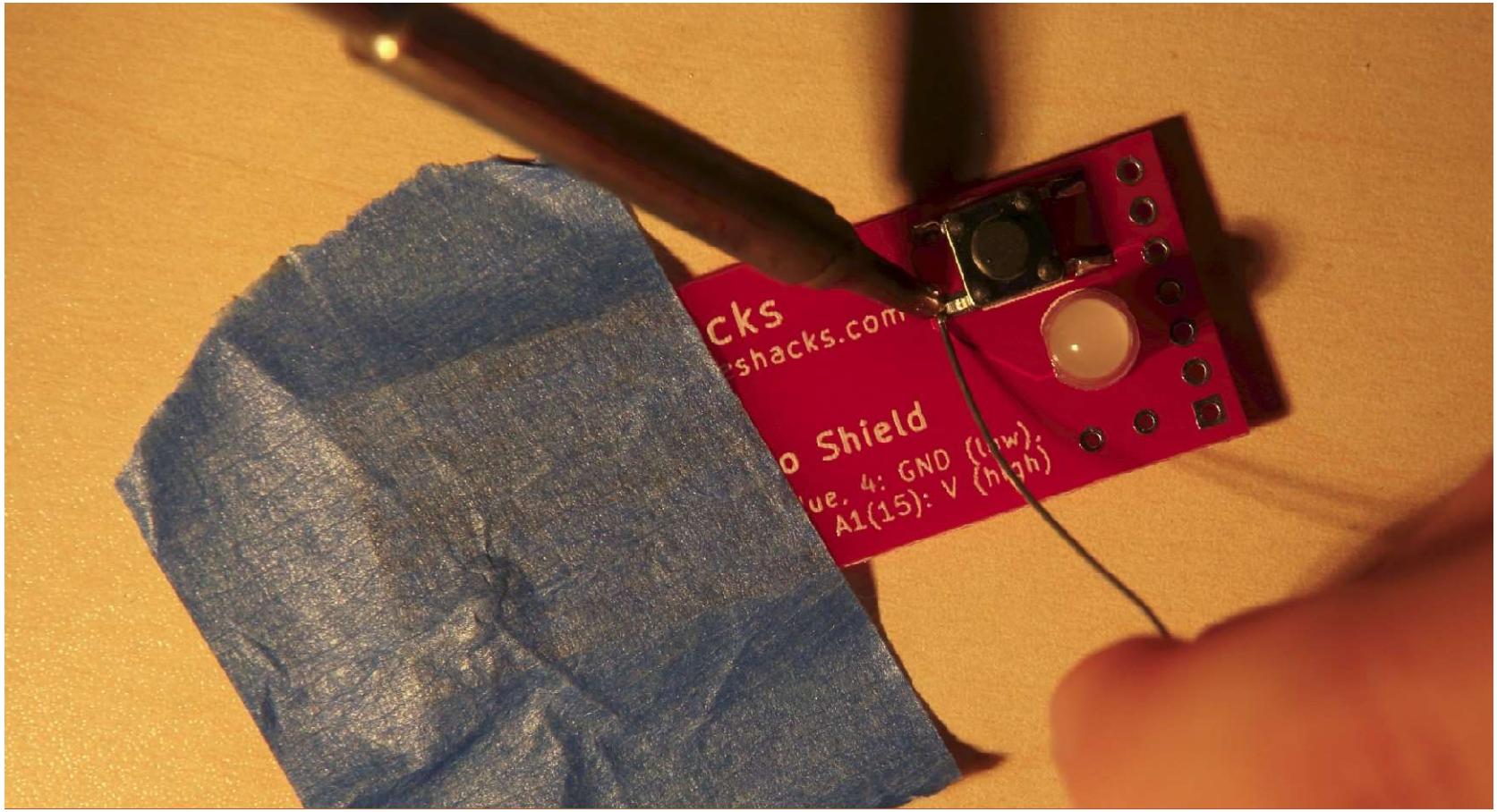
Place a Small Amount of Solder on One of the Button Pads



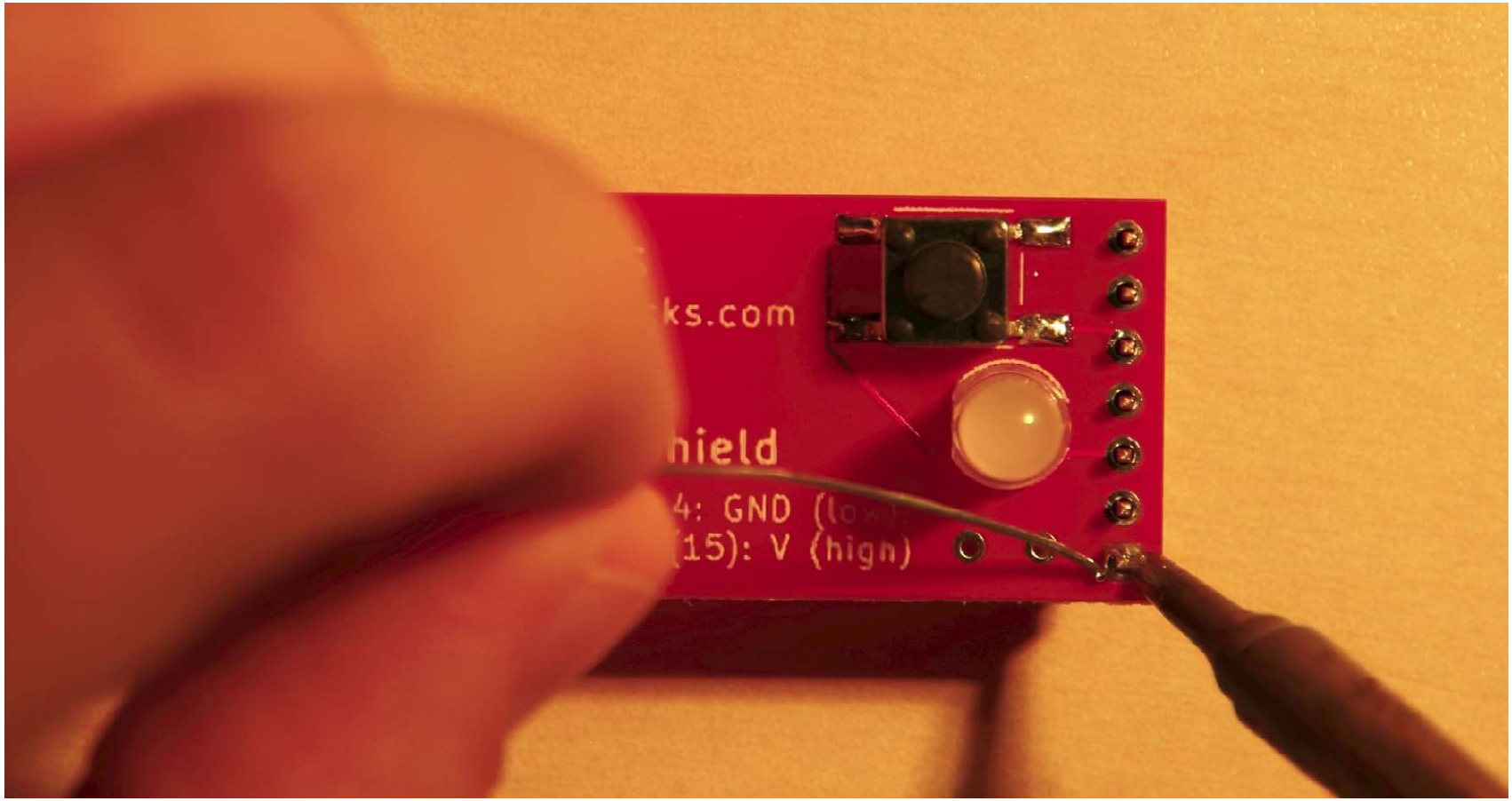
Solder the Button into Place by Remelting the Soldered Pad



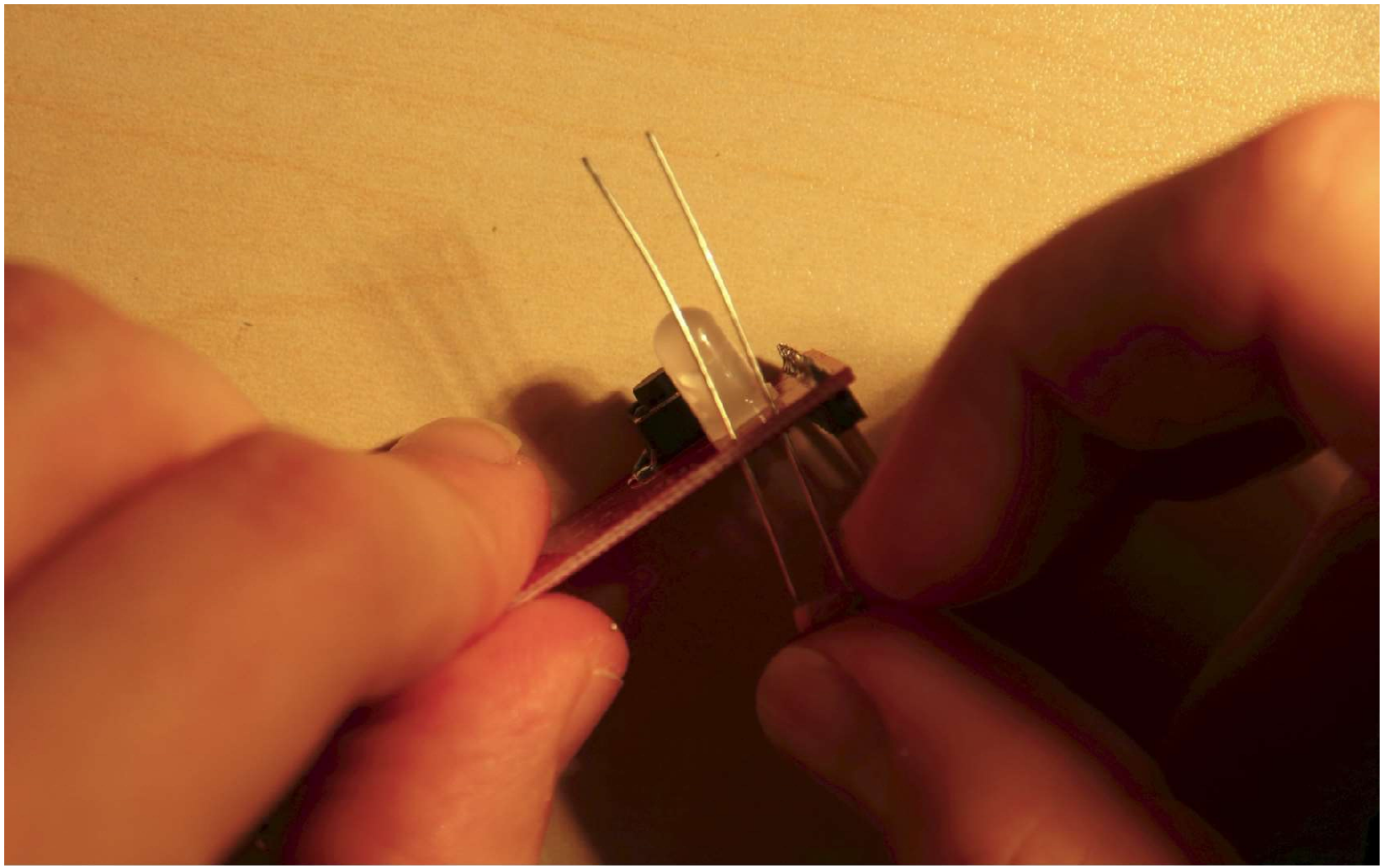
Solder the Other Button Pins



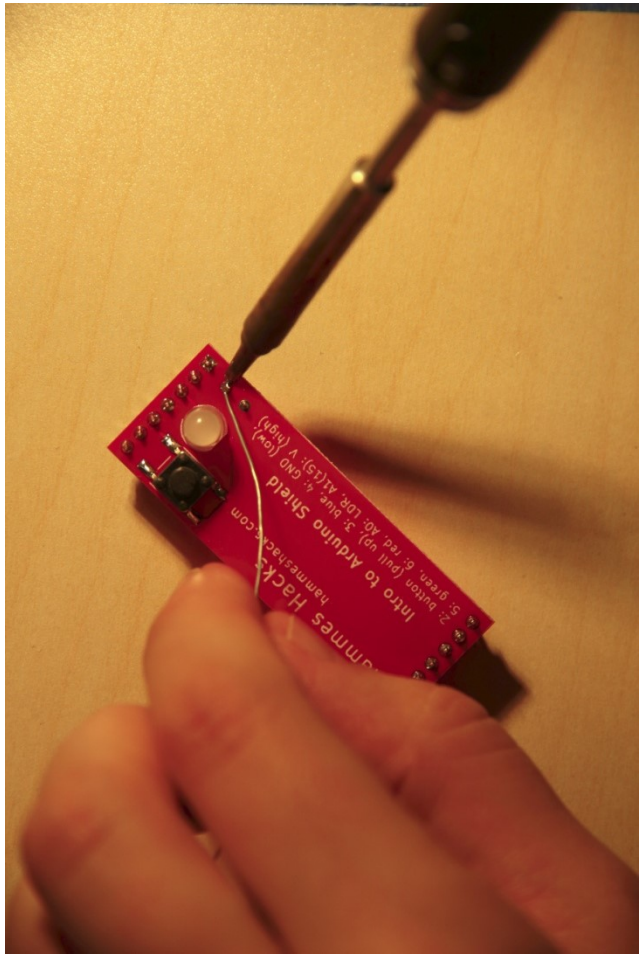
Solder the Pin Header



Place the LDR into the Board and Trim the Leads



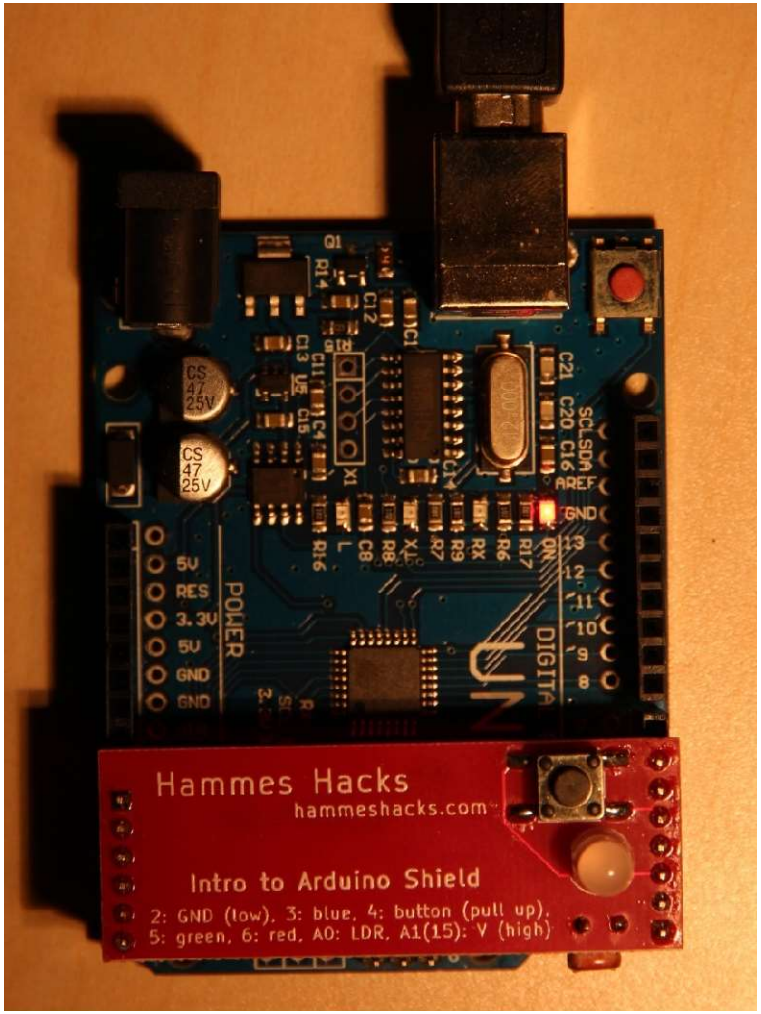
Solder the LDR



Bend the LDR Up

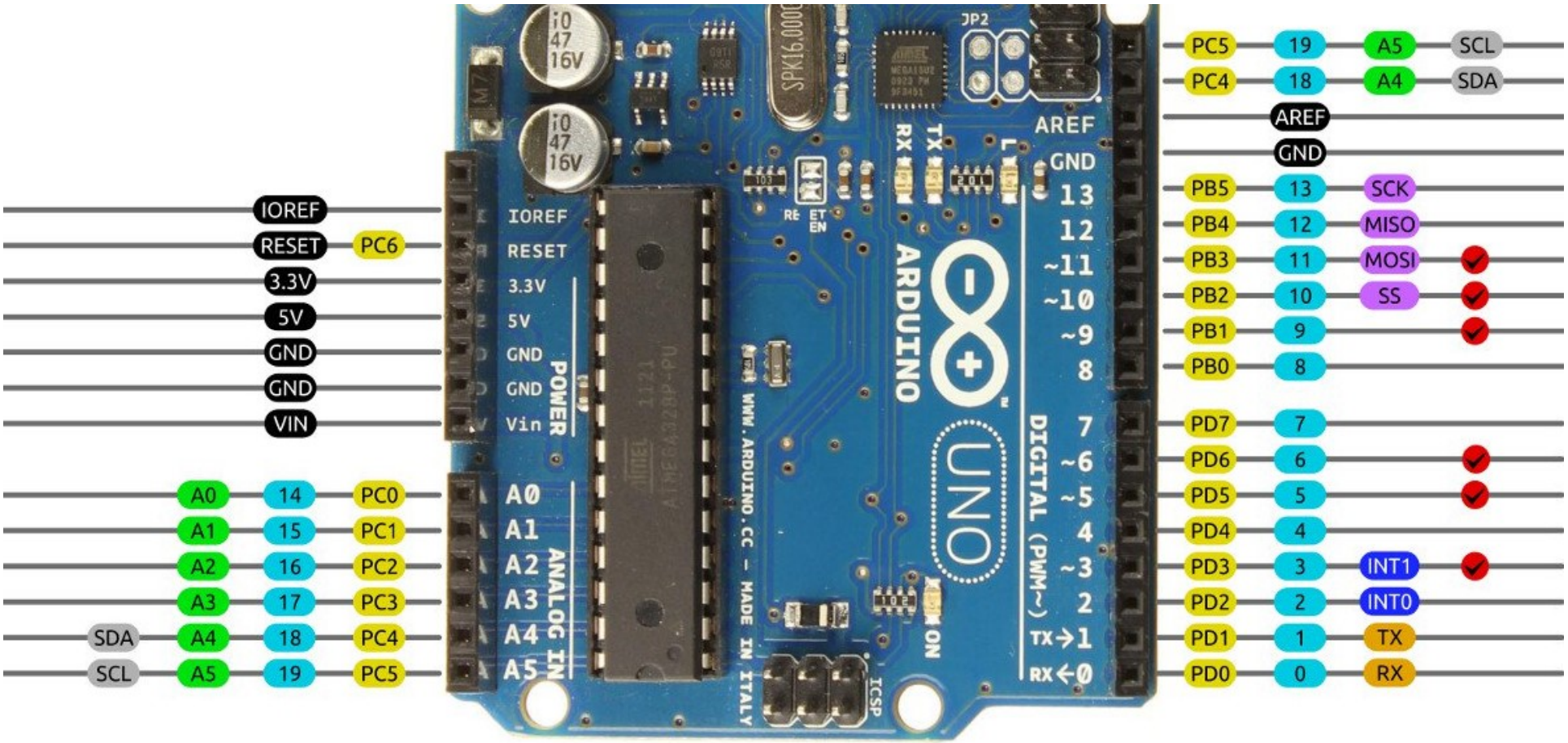
Test the Shield

- Plug the Shield into the Arduino
- Plug the Arduino into the computer (or power)
- The button should change the color of the LED and ambient light should change the LED brightness

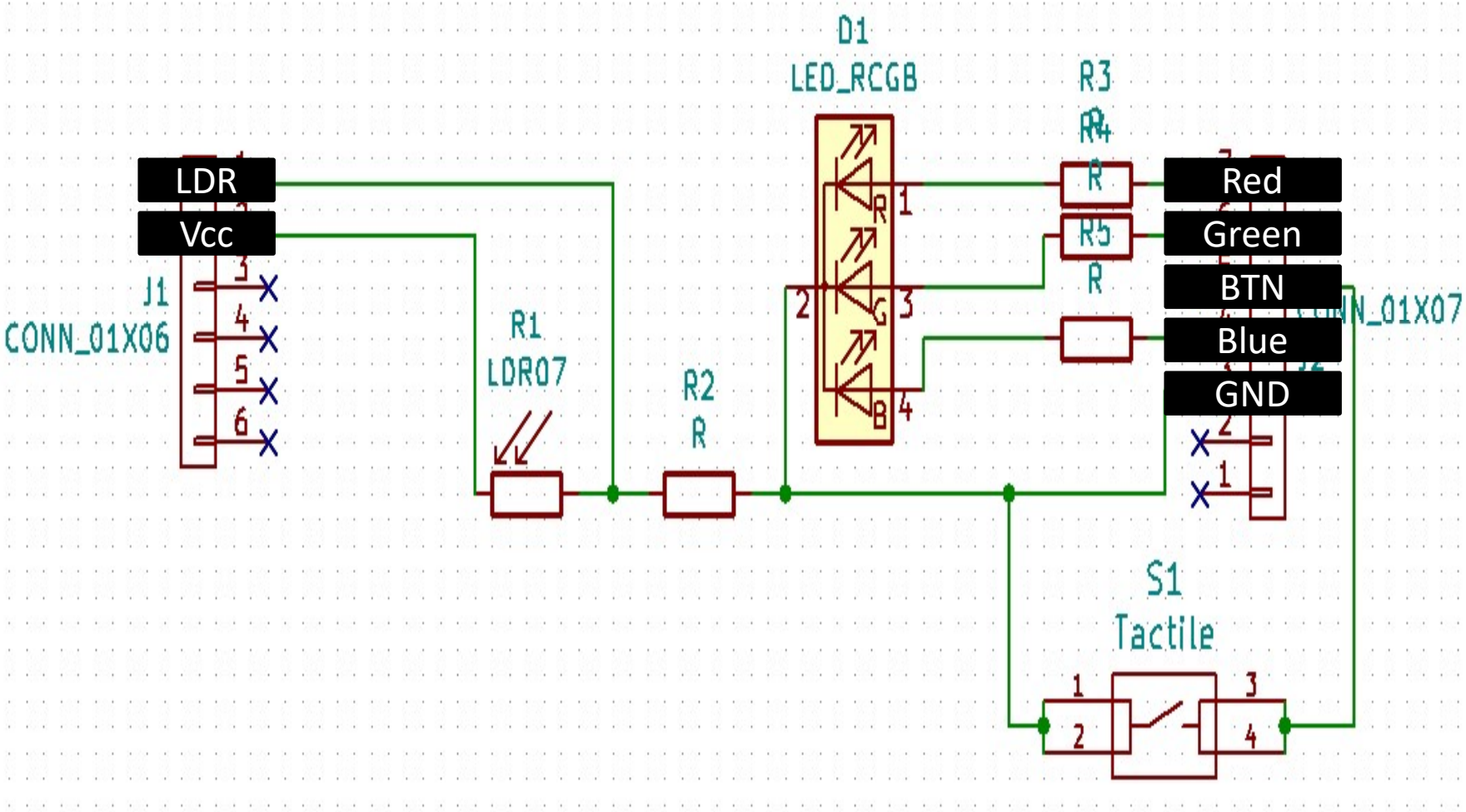


THE ELECTRONICS

Arduino Uno


IOREF
RESET
3.3V
5V
GND
GND
VIN
PC6
A0 14 PC0
A1 15 PC1
A2 16 PC2
A3 17 PC3
SDA A4 18 PC4
SCL A5 19 PC5
PC5 19 A5 SCL
PC4 18 A4 SDA
AREF
GND
PB5 13 SCK
PB4 12 MISO
PB3 11 MOSI
PB2 10 SS
PB1 9
PB0 8
PD7 7
PD6 6
PD5 5
PD4 4
PD3 3 INT1
PD2 2 INT0
PD1 1 TX
PD0 0 RX
POWER
ANALOG IN
DIGITAL (PWM~)
ARDUINO
UNO
WWW.ARDUINO.CC — MADE IN ITALY
ICSP
AVR
DIGITAL
ANALOG
POWER
SERIAL
SPI
I2C
PWM
INTERRUPT

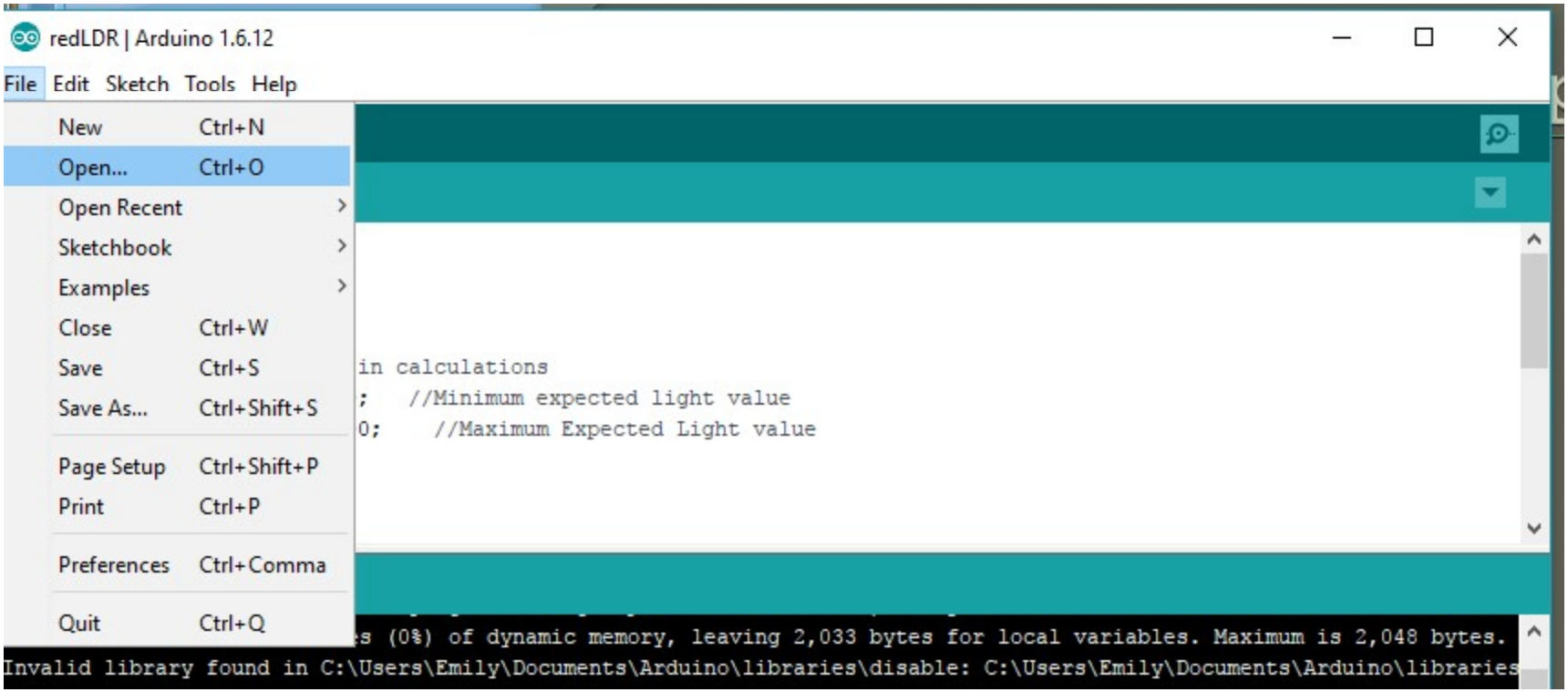
The Intro Shield



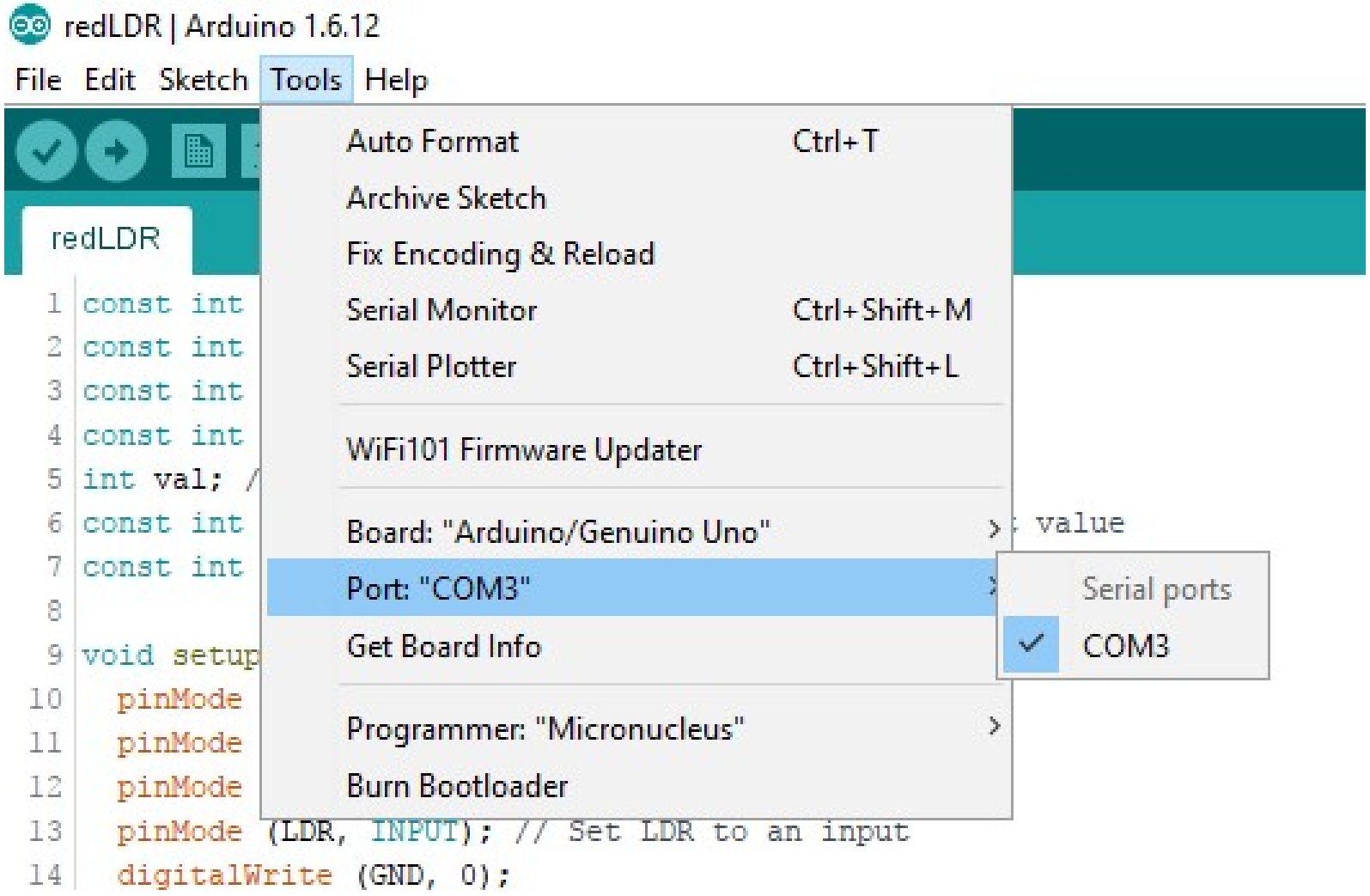
PROGRAMING

Setting Things Up

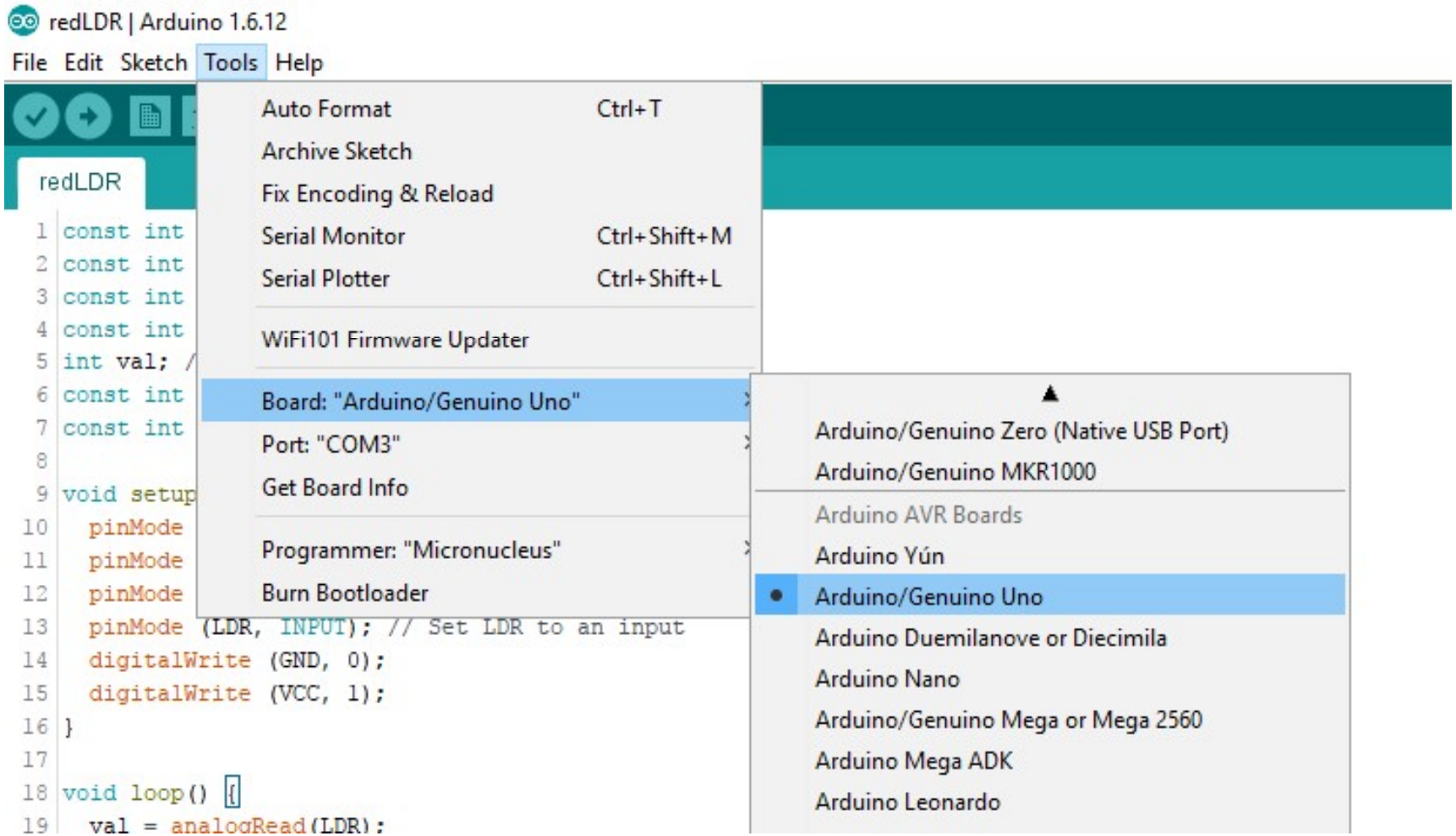
- Install the Arduino Software
- Open IntroToAdruinoShieldTest



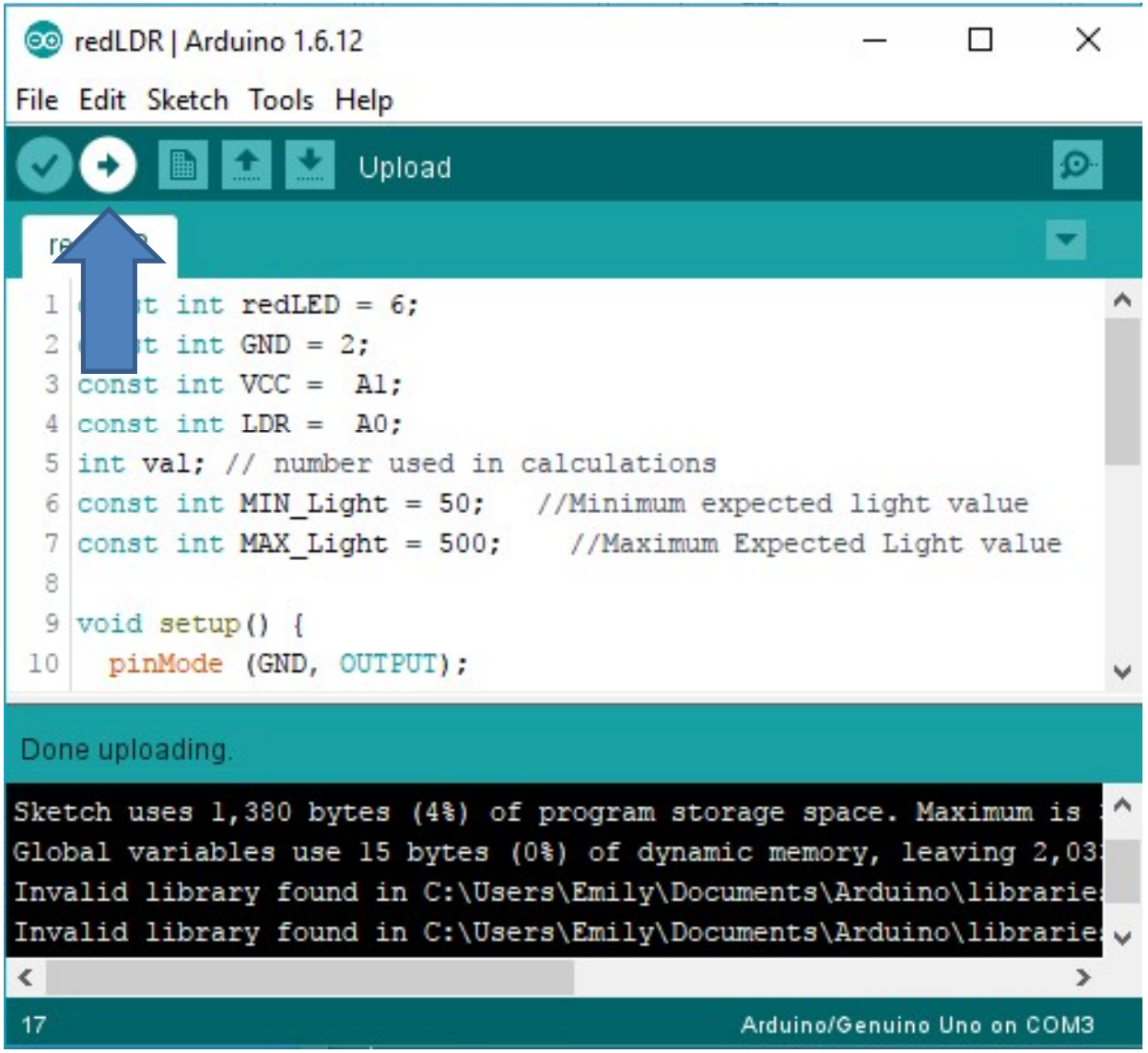
Selecting a ComPort



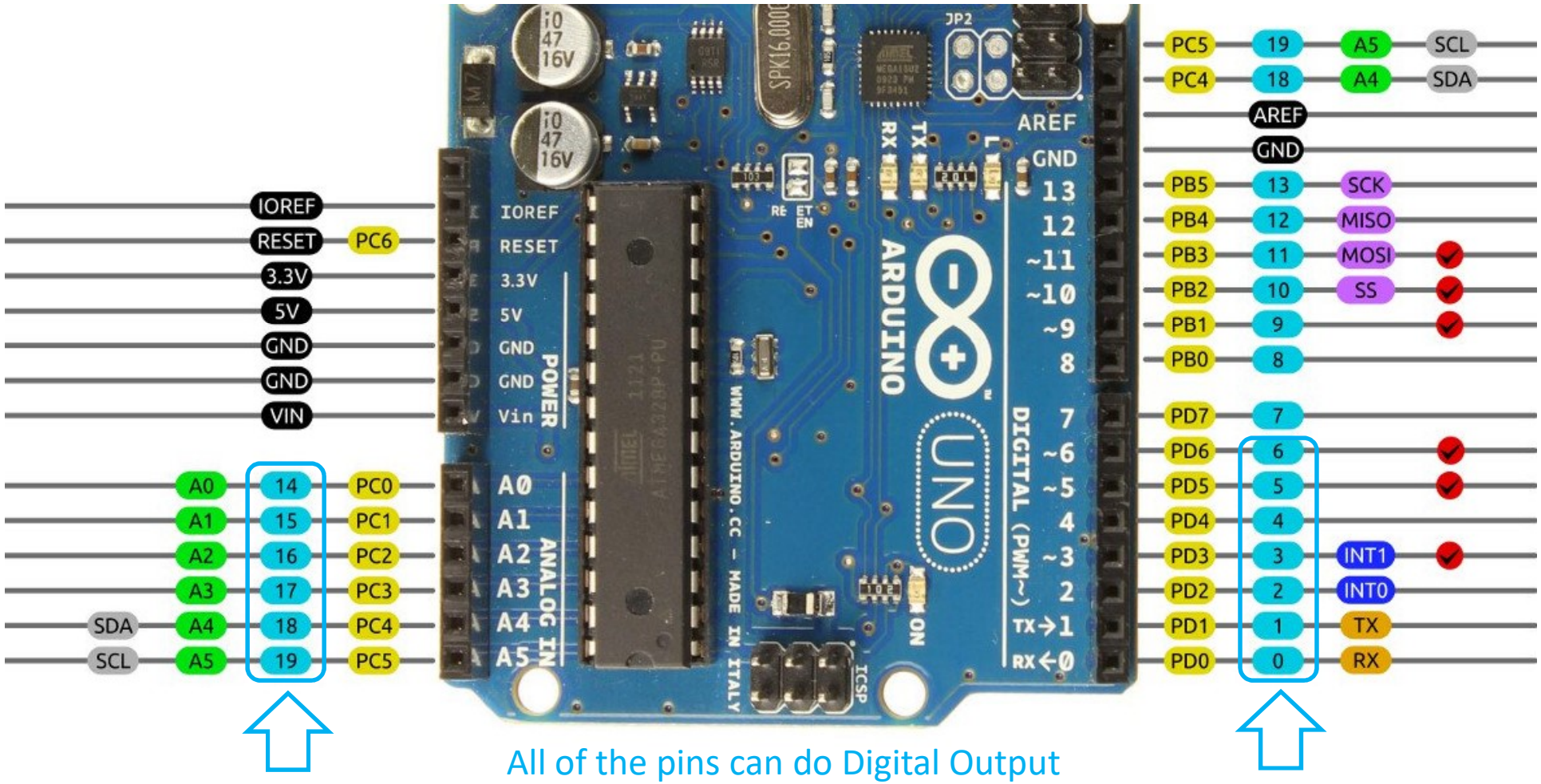
Selecting the Arduino



Uploading Code



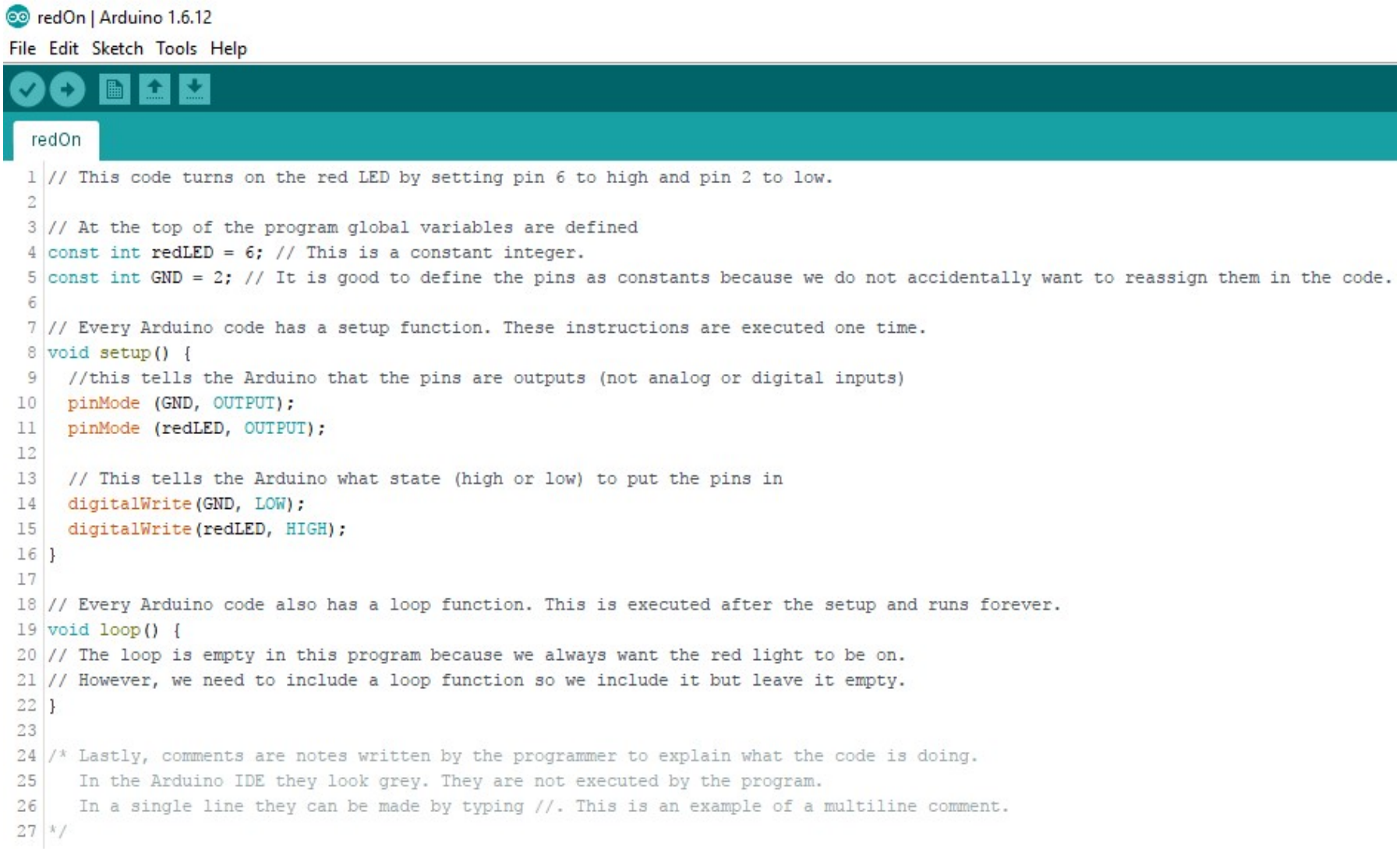
Arduino Uno



All of the pins can do Digital Output

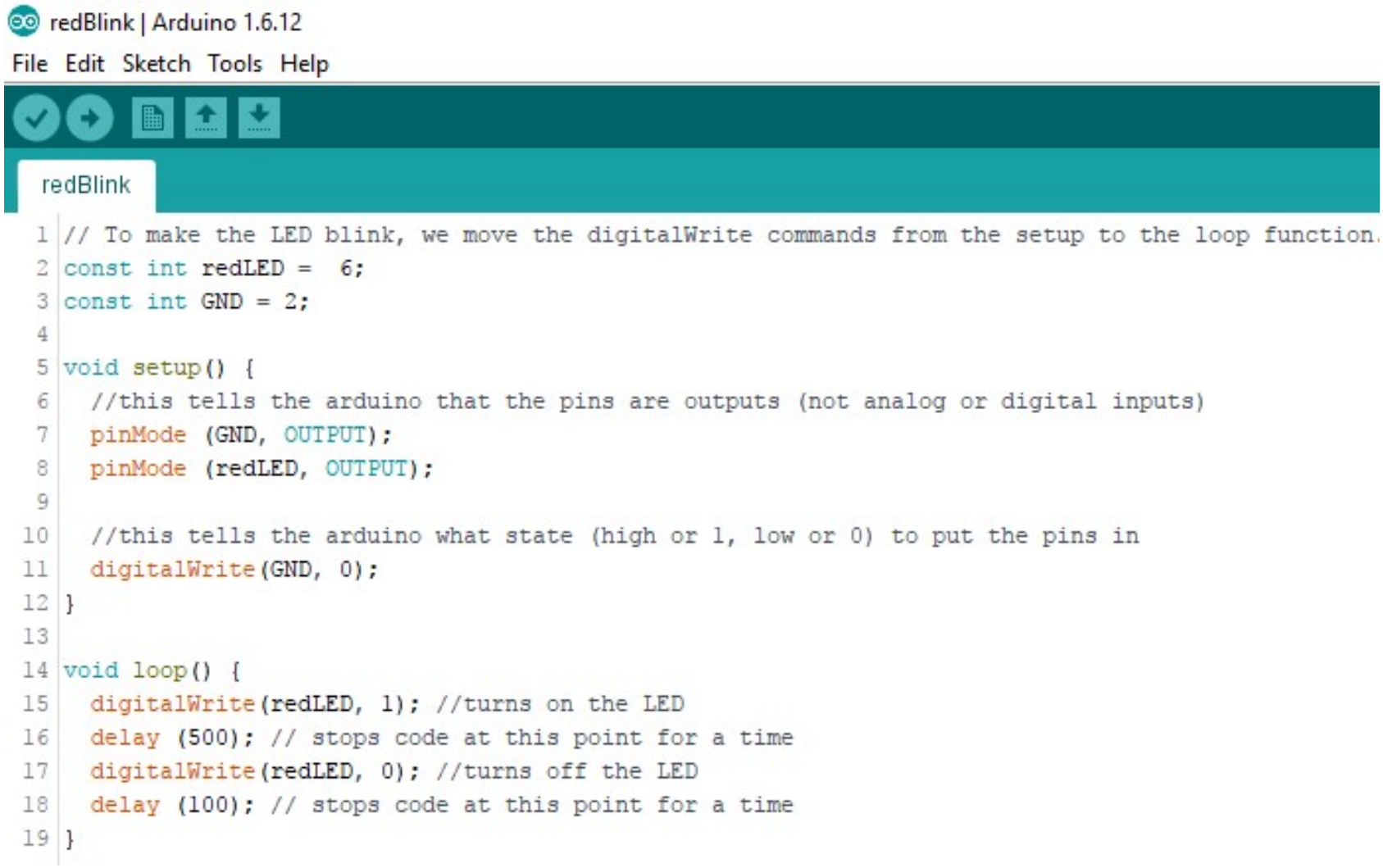
AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

RedOn



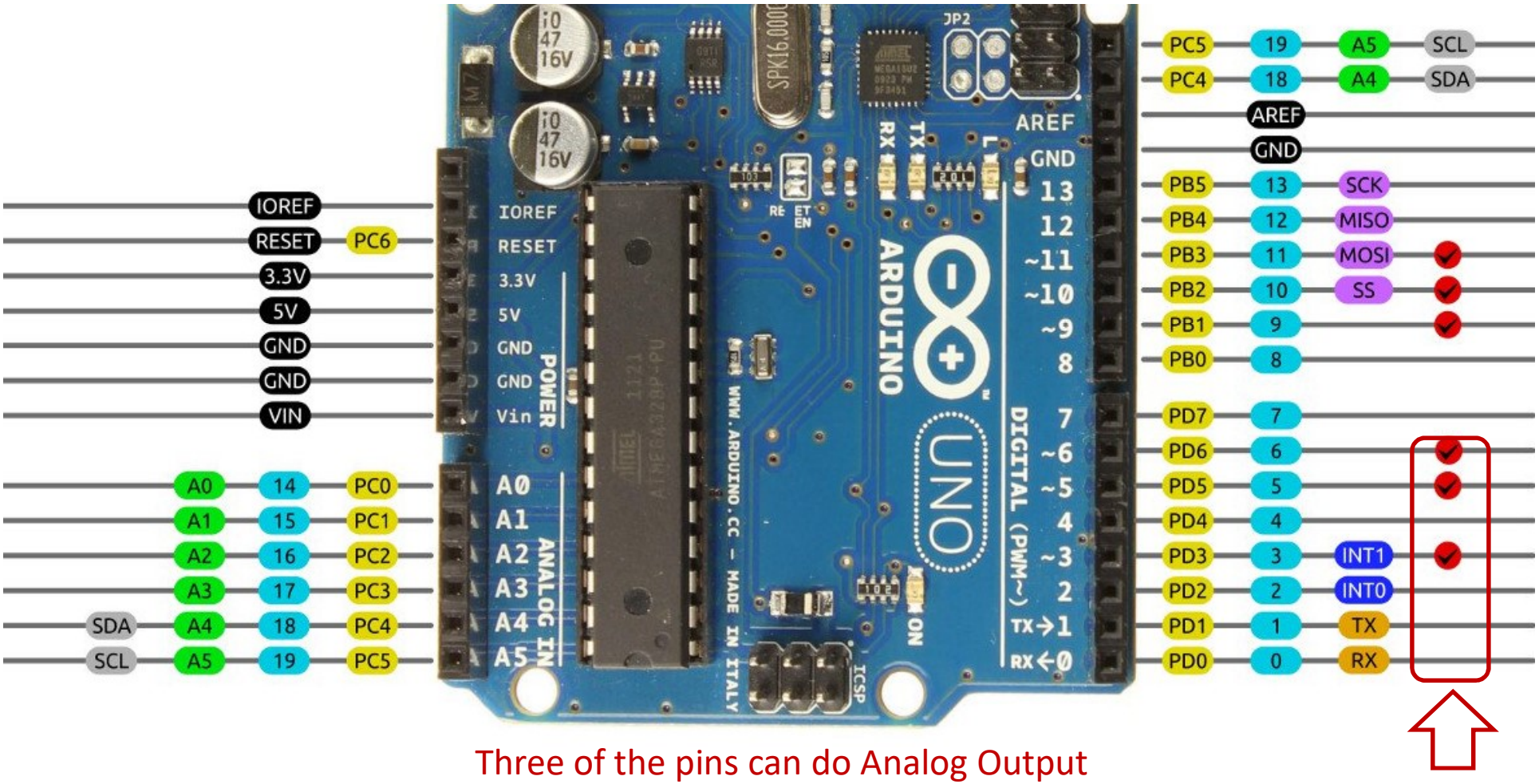

```
// This code turns on the red LED by setting pin 6 to high and pin 2 to low.

// At the top of the program global variables are defined
const int redLED = 6; // This is a constant integer.
const int GND = 2; // It is good to define the pins as constants because we do not accidentally want to reassign them in the code.

// Every Arduino code has a setup function. These instructions are executed one time.
void setup() {
  //this tells the Arduino that the pins are outputs (not analog or digital inputs)
  pinMode (GND, OUTPUT);
  pinMode (redLED, OUTPUT);

  // This tells the Arduino what state (high or low) to put the pins in
  digitalWrite(GND, LOW);
  digitalWrite(redLED, HIGH);
}

// Every Arduino code also has a loop function. This is executed after the setup and runs forever.
void loop() {
// The loop is empty in this program because we always want the red light to be on.
// However, we need to include a loop function so we include it but leave it empty.
}

/* Lastly, comments are notes written by the programmer to explain what the code is doing.
   In the Arduino IDE they look grey. They are not executed by the program.
   In a single line they can be made by typing //. This is an example of a multiline comment.
*/
```

RedBlink




```
1  // To make the LED blink, we move the digitalWrite commands from the setup to the loop function.
2  const int redLED =  6;
3  const int GND = 2;
4
5  void setup() {
6    //this tells the arduino that the pins are outputs (not analog or digital inputs)
7    pinMode (GND, OUTPUT);
8    pinMode (redLED, OUTPUT);
9
10   //this tells the arduino what state (high or 1, low or 0) to put the pins in
11   digitalWrite(GND, 0);
12 }
13
14 void loop() {
15   digitalWrite(redLED, 1); //turns on the LED
16   delay (500); // stops code at this point for a time
17   digitalWrite(redLED, 0); //turns off the LED
18   delay (100); // stops code at this point for a time
19 }
```

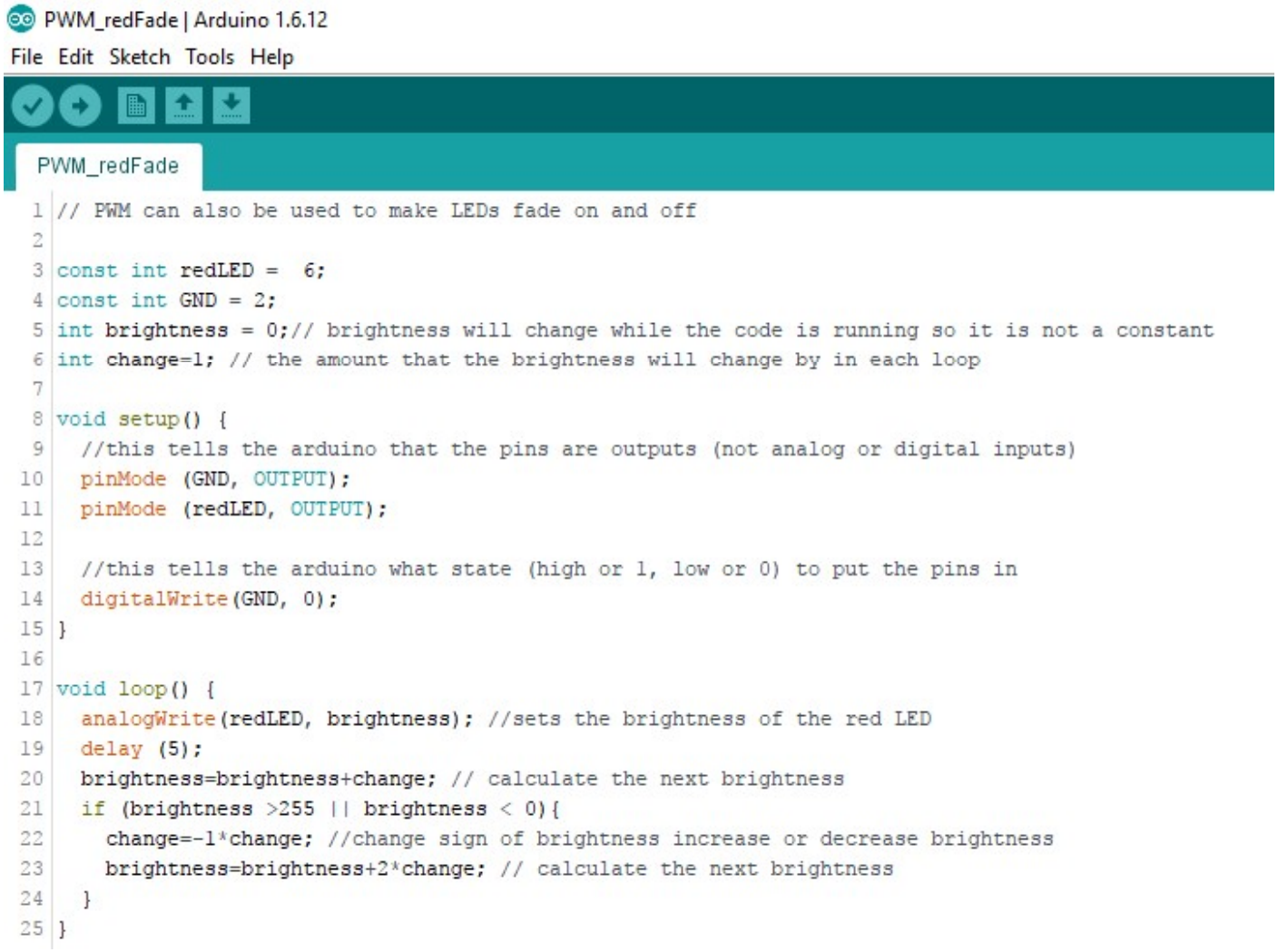
Arduino Uno



Three of the pins can do Analog Output

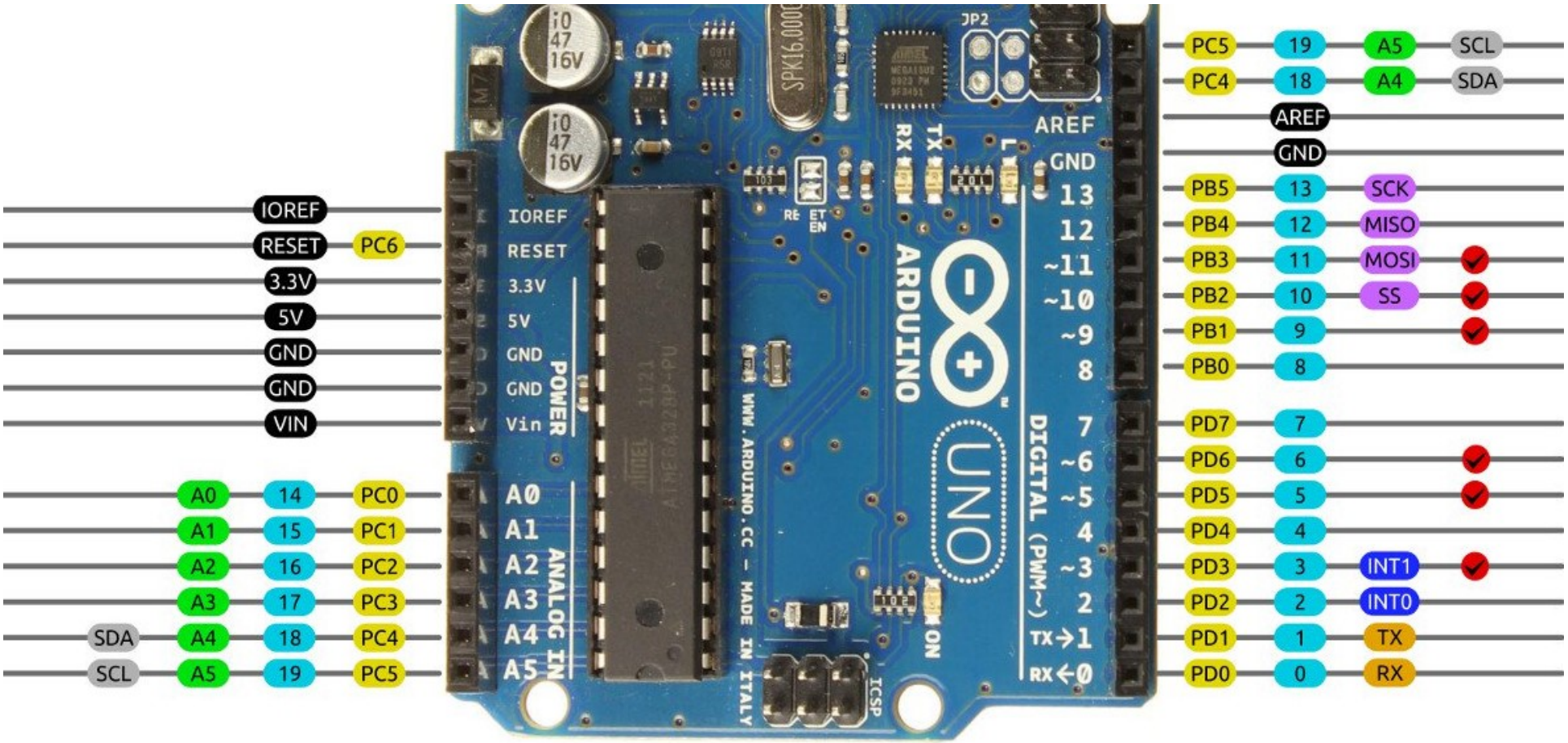
AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

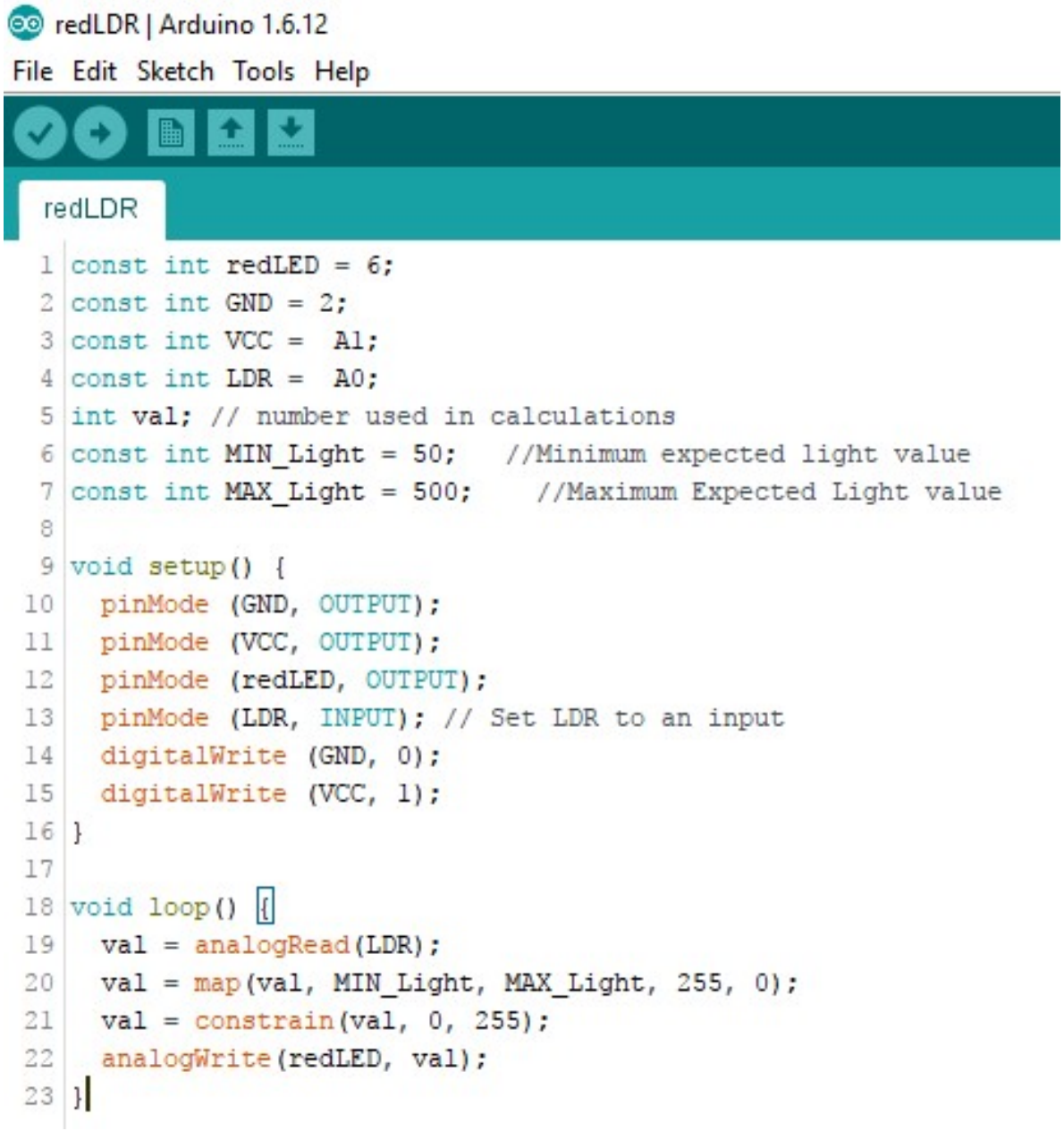
PWM_redFade



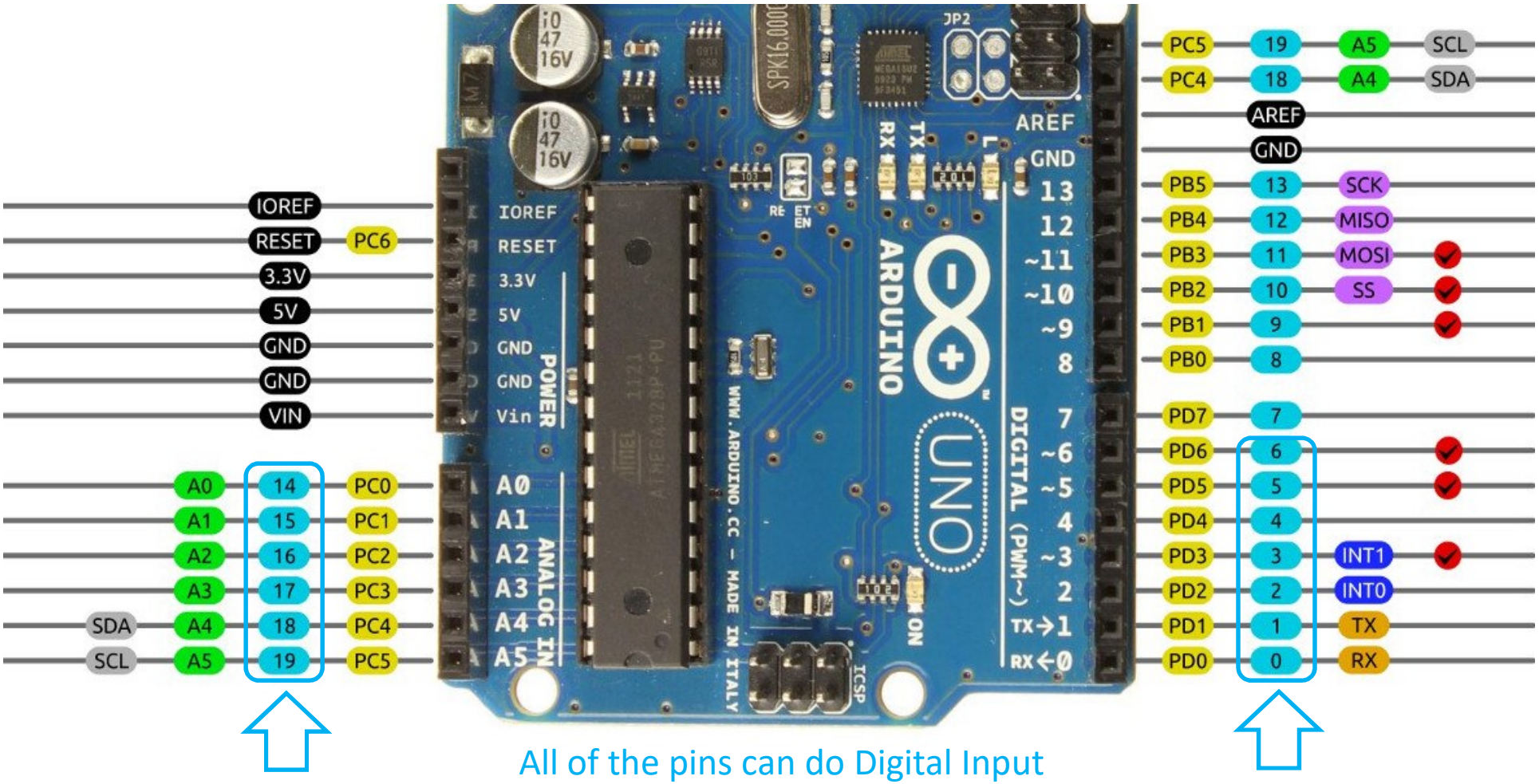

```
// PWM can also be used to make LEDs fade on and off

const int redLED =  6;
const int GND = 2;
int brightness = 0;// brightness will change while the code is running so it is not a constant
int change=1; // the amount that the brightness will change by in each loop

void setup() {
  //this tells the arduino that the pins are outputs (not analog or digital inputs)
  pinMode (GND, OUTPUT);
  pinMode (redLED, OUTPUT);

  //this tells the arduino what state (high or 1, low or 0) to put the pins in
  digitalWrite(GND, 0);
}

void loop() {
  analogWrite(redLED, brightness); //sets the brightness of the red LED
  delay (5);
  brightness=brightness+change; // calculate the next brightness
  if (brightness >255 || brightness < 0){
    change=-1*change; //change sign of brightness increase or decrease brightness
    brightness=brightness+2*change; // calculate the next brightness
  }
}
```

Arduino Uno


IOREF
RESET
3.3V
5V
GND
GND
VIN
PC6
A0 14 PC0
A1 15 PC1
A2 16 PC2
A3 17 PC3
SDA A4 18 PC4
SCL A5 19 PC5
PC5 19 A5 SCL
PC4 18 A4 SDA
AREF
GND
PB5 13 SCK
PB4 12 MISO
PB3 11 MOSI
PB2 10 SS
PB1 9
PB0 8
PD7 7
PD6 6
PD5 5
PD4 4
PD3 3 INT1
PD2 2 INT0
PD1 1 TX
PD0 0 RX
POWER
ANALOG IN
DIGITAL (PWM~)
ARDUINO
UNO
WWW.ARDUINO.CC — MADE IN ITALY
ICSP
AVR
DIGITAL
ANALOG
POWER
SERIAL
SPI
I2C
PWM
INTERRUPT


redLDR | Arduino 1.6.12
File Edit Sketch Tools Help
redLDR
const int redLED = 6;
const int GND = 2;
const int VCC = A1;
const int LDR = A0;
int val; // number used in calculations
const int MIN_Light = 50; //Minimum expected light value
const int MAX_Light = 500; //Maximum Expected Light value
void setup() {
pinMode (GND, OUTPUT);
pinMode (VCC, OUTPUT);
pinMode (redLED, OUTPUT);
pinMode (LDR, INPUT); // Set LDR to an input
digitalWrite (GND, 0);
digitalWrite (VCC, 1);
}
void loop() {
val = analogRead(LDR);
val = map(val, MIN_Light, MAX_Light, 255, 0);
val = constrain(val, 0, 255);
analogWrite(redLED, val);
}

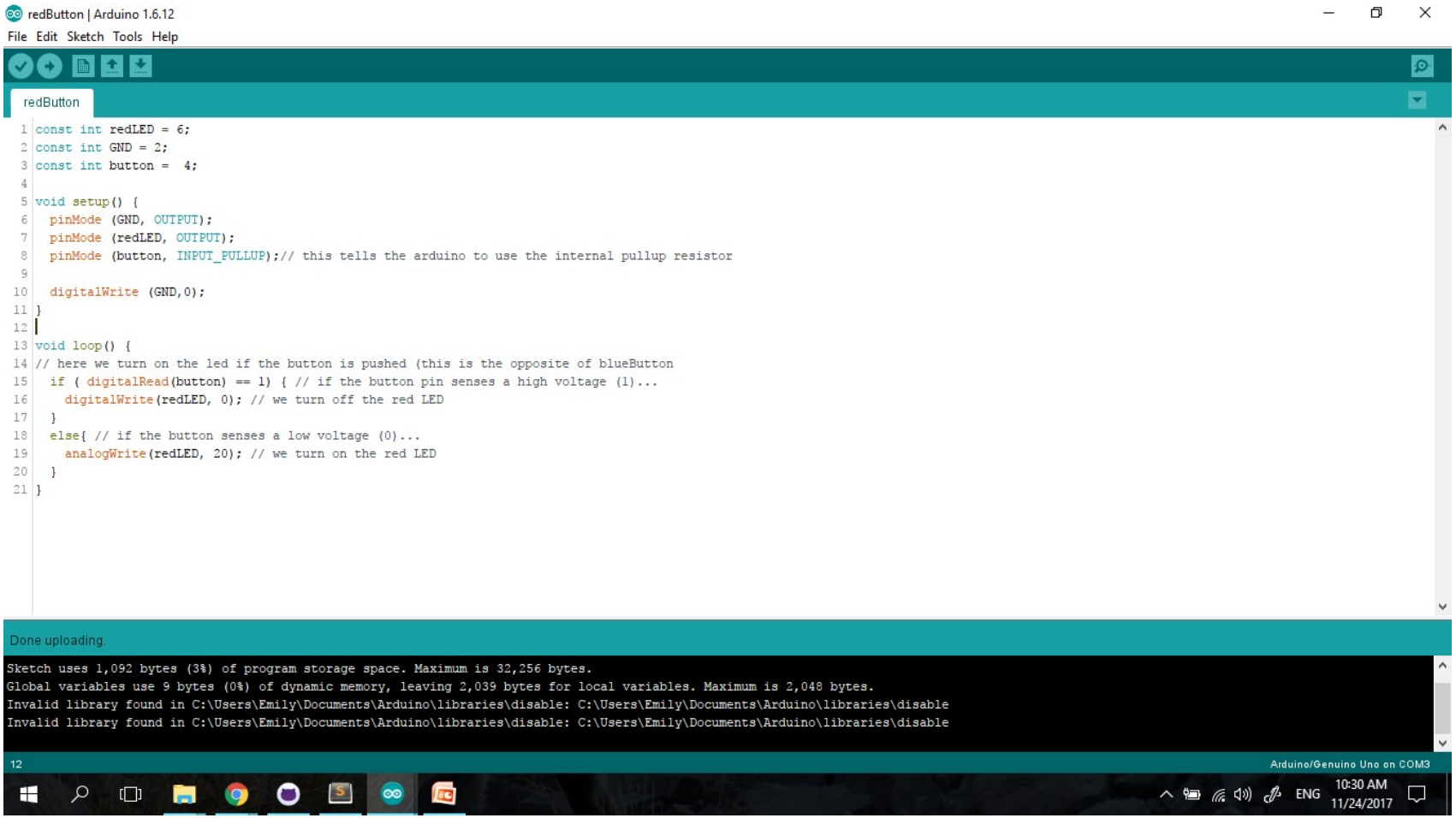
Arduino Uno



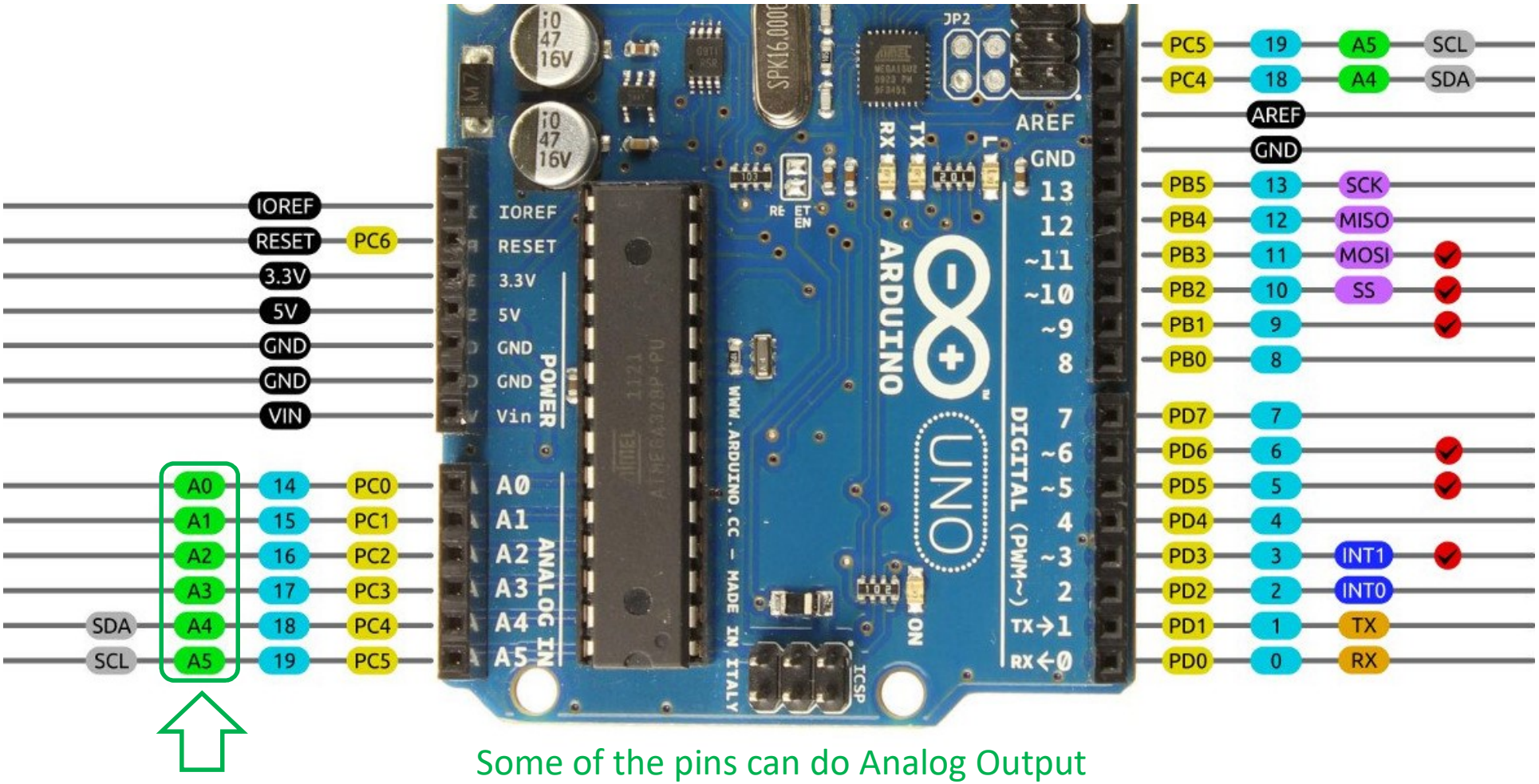
All of the pins can do Digital Input

AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

redButton


redButton | Arduino 1.6.12
File Edit Sketch Tools Help
redButton
const int redLED = 6;
const int GND = 2;
const int button = 4;
void setup() {
pinMode (GND, OUTPUT);
pinMode (redLED, OUTPUT);
pinMode (button, INPUT_PULLUP);// this tells the arduino to use the internal pullup resistor
digitalWrite (GND,0);
}
void loop() {
// here we turn on the led if the button is pushed (this is the opposite of blueButton
if (digitalRead(button) == 1) { // if the button pin senses a high voltage (1)...
digitalWrite(redLED, 0); // we turn off the red LED
}
else{ // if the button senses a low voltage (0)...
analogWrite(redLED, 20); // we turn on the red LED
}
}
Done uploading.
Sketch uses 1,092 bytes (3%) of program storage space. Maximum is 32,256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for local variables. Maximum is 2,048 bytes.
Invalid library found in C:\Users\Emily\Documents\Arduino\libraries\disable: C:\Users\Emily\Documents\Arduino\libraries\disable
Invalid library found in C:\Users\Emily\Documents\Arduino\libraries\disable: C:\Users\Emily\Documents\Arduino\libraries\disable
12
Arduino/Genuino Uno on COM3
ENG
10:30 AM
11/24/2017

Arduino Uno



Some of the pins can do Analog Output

AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Nonblocking State Machine

statemachiene | Arduino 1.6.12

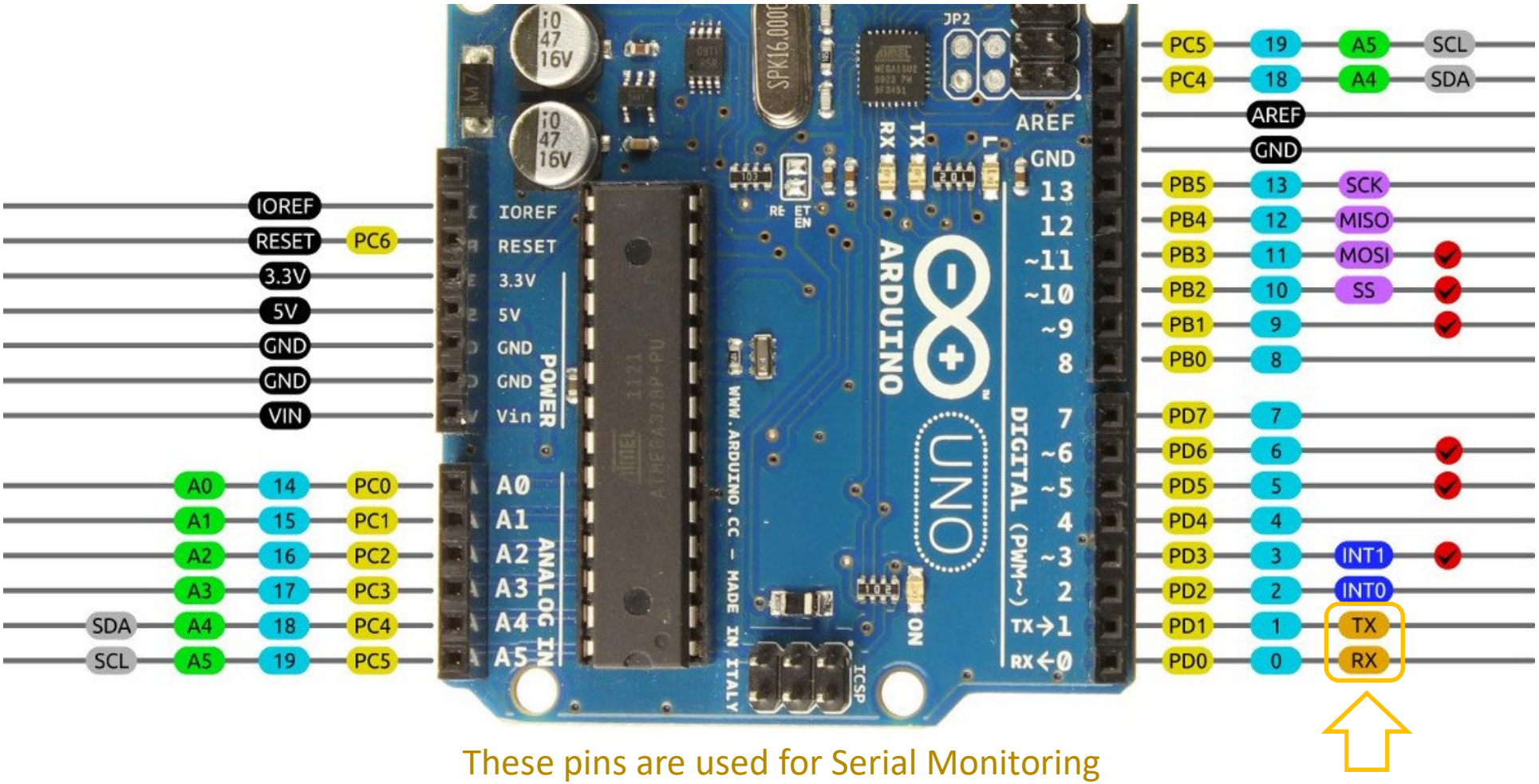
File Edit Sketch Tools Help

statemachiene

```
1  //If you need the arduino to do two things but y
2  //Here red blinks at an interval of rTime and bl
3
4  const int redLED = 6;
5  const int blueLED = 3;
6  const int GND = 2;
7  const int rTime=800;
8  const int bTime=650;
9  unsigned long redTime = millis();
10 unsigned long blueTime = millis();
11
12 void setup() {
13   Serial.begin(9600);
14   pinMode (GND, OUTPUT);
15   pinMode (redLED, OUTPUT);
16   pinMode (blueLED, OUTPUT);
17   digitalWrite (GND,0);
18 }
```

```
20 void blueOn() {
21     analogWrite(blueLED, 20); // we turn on the
22     analogWrite(redLED, 0); // we turn off the r
23     Serial.println(redTime);
24   }
25
26 void redOn() {
27     analogWrite(blueLED, 0); // we turn off the
28     analogWrite(redLED, 20); // we turn on the r
29     Serial.println(blueTime);
30   }
31
32 void loop()
33 // Here we make a state macheine. All it does is
34 if (millis() >redTime){
35  redOn();
36  redTime=millis()+rTime;
37 }
38
39 if (millis() > blueTime){
40   blueOn();
41   blueTime=millis()+bTime;
42 }
43 }
```

Arduino Uno



These pins are used for Serial Monitoring

AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT