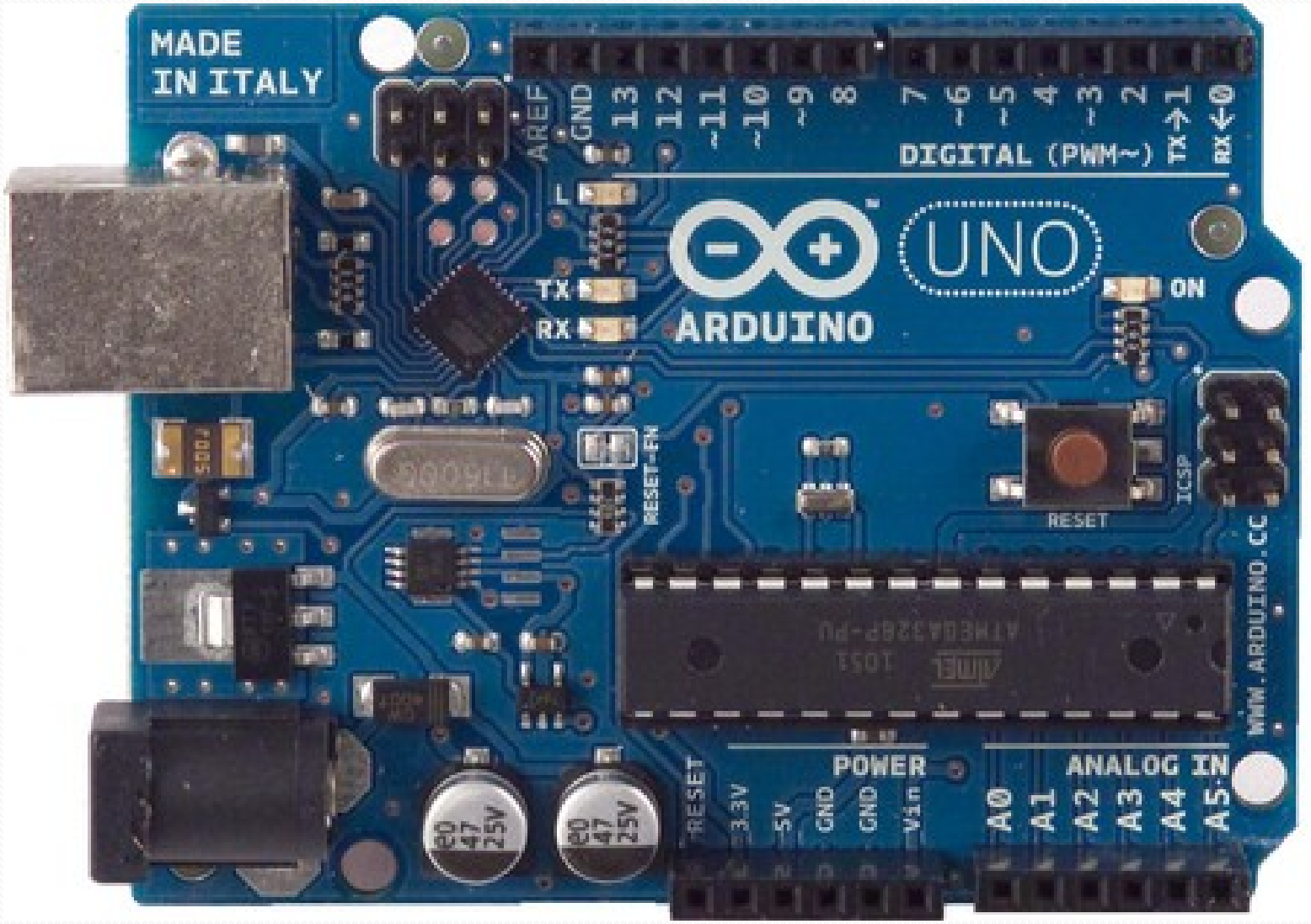
Line Following (Using Arduino)

By: Suvansh Kumar
Vidushi Daddu

Arduino UNO

- For small projects we use Arduino UNO board.
- The MCU : ATmega328
- It has 6 analog input pins (Single purpose pins).
- 6 PWM pins
- Inbuilt programmer
- Needs an external motor driver
- No ports for LCD
- Inbuilt LED at pin No. 13


MADE
IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
RESET-EN
RESET
ICSP
WWW.ARDUINO.CC
POWER
ANALOG IN
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Arduino Software

- It has many inbuilt libraries.
- Has a serial monitor which can be used to print values instead of an LCD
- Single software for both compiling and writing the code into the chip.

Steps To Use The Software

1. Choose the appropriate board and port on which the MCU is connected.
2. Go to Tools->board for changing the board.
3. Tools->Port for changing the port.
4. Write the code.
5. Upload it.

Arduino Programming

- The code consists of three parts:

1. Including the libraries and defining the variables.
2. void Setup() function: For the tasks which needs to be performed only in the starting of the code.
3. void loop() function: For the tasks that need to be performed after the execution of setup(). This is an infinite loop.

Input And Output Pins

- We have to first define a pin as input or output.
- We only need to define the pins we need to use as digital pins.
- Function: pinMode(pinNumber,OUTPUT/INPUT)
- Pins we use for pwm output (analog output) need not be defined as output.
- Analog input pins need not be defined.

Functions:

- digitalRead(pinNumber) : To take digital input.
- digitalWrite(pinNumber, HIGH/LOW): To give digital output.
- analogRead(pinNumber) : To get Analog input.
- analogWrite(pinNumber, 0-255) : To give pwm.
- delay(milliseconds) : To give a delay in milliseconds.
- constrain(x, a, b) : It is
  = x if a < x < b
  = a if x < a
  = b if x > b

Using Serial Monitor

- First we need to initiate the data transfer.
- Function: Serial.begin(baudRate)
- Baud Rate: The rate at which the data is transferred. General value: 9600.
- To print a value on monitor:
  Serial.println(x).
  x: Any value or string.

A sample program

```
int l3, l2, l1, r1, r2, r3;
int motor11 = 10, motor12 = 12, motor21 = 11, motor22 = 13;
float pwm1 = 150, pwm2 = 150;
float kp=0, kd=0, ki=0, crr=0, ierr=0, err=0, derr=0, perr=0;
float sum=0, sum1=0, avg=0, avg1=0, pwm1=0, pwm2=0;

void setup(){
  kp=10;
  kd=100;
  pinMode(motor12,OUTPUT);
  pinMode(motor22,OUTPUT);
  Serial.begin(9600);
  avg1 = 3 ;
}

void loop(){
  int l3 = analogRead(A0);
  int l2 = analogRead(A1);
  int l1 = analogRead(A2);
  int r1 = analogRead(A3);
  int r2 = analogRead(A4);
  int r3 = analogRead(A5);
  sum = l3 + l2 + l1 + r1 + r2 + r3 ;
  avg = (l3*1 + l2*2 + l1*3 + r1*4 + r2*5 + r3*6)/sum ;
```

```
void loop(){
  int l3 = analogRead(A0);
  int l2 = analogRead(A1);
  int l1 = analogRead(A2);
  int r1 = analogRead(A3);
  int r2 = analogRead(A4);
  int r3 = analogRead(A5);
  sum = l3 + l2 + l1 + r1 + r2 + r3 ;
  avg = (l3*1 + l2*2 + l1*3 + r1*4 + r2*5 + r3*6)/sum ;
  if ( sum > 3600 )
    err = perr;
  else
    err = avg - avgl;
    derr = err - perr;
    ierr = ierr + err;
    crr = kp*err + kd*derr;
    pwm1 = pwm1 + crr;
    pwm2 = pwm2 - crr;
    pwm1 = constrain(pwm1,0,255);
    pwm2 = constrain(pwm2,0,255);
    perr = err;
    digitalWrite(motorl2,LOW);
    digitalWrite(motor22,LOW);
    analogWrite(motorl1,pwm1);
    analogWrite(motor21,pwm2);
}
```

Understanding The Effect Of Kp And Kd values

- Derr value is always small. (0.00-0.02)
- It doesn’t depend on the position of the robot i.e. does not depend on how much the robot has deflected or how much the error is.
- It just depends on the difference between previous and current state which is always small due to high frequency.
- So Kd decides the constant amount of change that should be given to the pwm when the robot deflects.

- Err value depends upon the position of the robot i.e. the amount of deflection.
- So, when the err is more, the contribution of the kp term is more unlike kd which remains constant.
- In the correction term, when the robot is on the line, kd contributes more than the kp as err is close to zero.
- When the robot is out of line, derr is zero so its kp which contributes more.
- So, when line is straight, kp value will be less as compared to when the line is curved.

Steps To Set Kp And Kd

- First set both to zero.
- Increase the value of Kp by 1 until it start to wobble around the line. Once it is following the line till a reasonable distance, stop increasing.
- Now start increasing the kd value by 10 until the wobbling is reduced to almost negligible.
- Kp and Kd values will be different for curved and straight paths.

Thank you