

The Working Principle Of An Arduino.

Yusuf Abdullahi Badamasi
Nigerian Turkish Nile University
Abuja, Nigeria
Email:yusufbadamasi23@gmail.com

Abstract—In this paper, we analyze the working principle of an arduino. These days many people try to use the arduino because it makes things easier due to the simplified version of C++ and the already made Arduino microcontroller(atmega328 microcontroller [1]) that you can programme,erase and reprogramme at any given time .In this paper we will discuss the hardware components used in the arduino board, the software used to programme it(Arduino board) with the guide on how to write and construct your own projects, and a couple of examples of an arduino project,This will give you the overall view of an arduino uno, that after reading this paper you will get the basic concept and use of an arduino uno.

Keywords—Open-source platform, Hardware ,Software, Micro-controller.

I. INTRODUCTION

Arduino is an open-source platform [2] used for constructing and programing of electronics.It can receive and send information to most devices, and even through the internet to cammand the specific electronic device.it uses a hardware called arduino uno [3] circuit board and software programme(Simplified C++ [1]) to programme the board.

In these modern day, Arduino are used alot in microcontroller [4] programing among other things due to its user friendly or easy to use setting, like any microcontoller an arduino is a circuit board with chip that can be programmed to do numerous number of tasks, it sends information from the computer programme to the Arduino microcontroller and finally to the specific circuit or machine with multiple circuits in order to execute the specific command.An arduino can help you read information from input devices [5] such as e.g Sensors, Antenna, Trimmer(potentiometer) e.t.c... and can also send information to output devices such as LED [5], Speakers, LCD Screen, DC motor e.t.c...

II. ARDUINO BOARD

The Arduino platform [6] has become well aquinted with people into electronics. Unlike most previous programmable circuit boards [7], the Arduino does not have a separate piece of hardware in order to load new code onto the board, you can simply use a USB cable to upload, and the software of the Arduino uses a simplified version of C++ [8], making it easier to learn to program,and it provides you with an easier environment that bypass the functions of the micro-controller [4] into a more accessible package.

An Arduino Board [7] can be classified into two parts :

- 1) Hardware: The Arduino board [7] hardware consist of many components that combine to make it work,

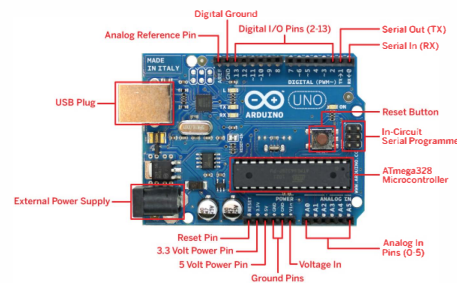


Fig. 1. Labelled Arduino Board

but we are going to discuss the main component on the board such as follows:

- USB Plug: This is the first part of the arduino because it is used to upload a programme to the microcontroller [4] and has a regulated power of 5volts which also power the Arduino board .
 - External Power Supply: This is only used to power the board and has a regulated voltage of 9 to 12 volts, mostly if the USB plug does not provide sufficient power for whatever you have programmed it to do.
 - Reset button: This button resets the arduino when it when its pressed incase you have uploaded another command and want the arduino to do it.
 - Microcontroller: This is the device that receive and send information or cammand to the respective circuit.
 - Analog Pins(0-5): This are analog input pins from A0 to A5.
 - Digital I/O Pins: This are the digital input, output Pins 2 to13.
 - In-Circuit Programmer : This is another source to upload or programme your programme,it can also be done using"TX-;1" output and "RX-;0" input.
 - Digital and analog Ground pins
 - Power Pins: we have 3.3 and 5 volts power pins e.t.c..
- 2) Software(The Arduino IDE [9]) : The software is a set of instructions that informs the hardware of what to do and how to do it. The Arduino IDE(Integrated Development Environment) is divided into three main parts:
 - a) Command Area: This is the area where you have the menu items such as File, Edit,

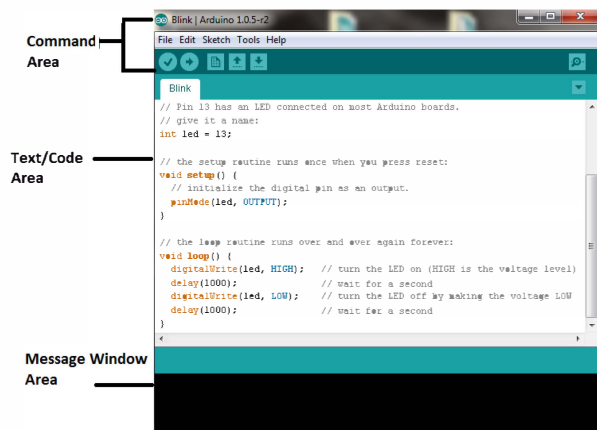


Fig. 2. Labelled IDE

Sketch, Tools, Help and Icons like Verify Icon for verification, Upload Icon for uploading your programme, New, Open, Save and Serial Monitor used for sending and receiving of data between the arduino and the IDE.

- b) Text Area: This is where you write your code which uses a simplified version of C++ programming language that makes it easier to write your programme, which is also called a sketch. When writing your code there are mainly two important parts :
- The setup function: Before the setup you need to initialize the variables you intend to use and assign them. Then the setup routine begins, This is where you set the initial condition of your variables and run preliminary code only once. Here is an example of how it should be written [10].

```
void setup() {
  This where you write
  your code which will run once.
}
```
 - Loop routine: This is the loop that runs or executes your main code over and over again. Here is an example..

```
void loop() {
  This is where your main code is
  written, to run repeatedly.
}
```
- c) Message Window Area: This shows message from the IDE in the black area, mostly on verification of your code.

III. APPLICATIONS(EXAMPLES)

In this part we take a look at some basic examples in arduino and notice how the application is written and sent to the arduino and resulting output. Here are some Application:

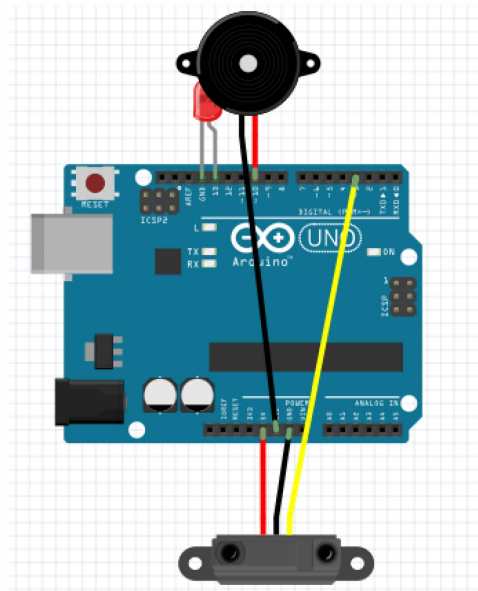


Fig. 3. Sensor Alarm

A. Sensor Alarm

Sensor alarm uses a sensor either infrared, motion any kind of sensor to detect movement then send the message to the arduino microcontroller which then turns ON or OFF the LED and speaker.

- COMPONENTS
 - 1) Arduino Uno
 - 2) LED
 - 3) Speaker
 - 4) Infrared Proximity Sensor
 - 5) Breadboard
- CIRCUIT [11]
- CODE [11]


```
int ledPin = 13; // choose the pin
                for the LED
int inputPin = 3; // choose the input
                pin (for PIR sensor)
int pirState = LOW; // we start,
                    assuming no motion
                    detected
int val = 0; // variable for reading
the pin status
int pinSpeaker = 10; //Set up a speaker
                    on a PWM pin
                    (digital 9, 10, or 11)
void setup() {
  pinMode(ledPin, OUTPUT); // declare LED
  as output
  pinMode(inputPin, INPUT); // declare
  sensor as input
  pinMode(pinSpeaker, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  val = digitalRead(inputPin); // read input
  value
```

```

if (val == HIGH) { // check if the input
is HIGH
digitalWrite(ledPin, HIGH); // turn LED
ON
playTone(300, 160);
delay(150);
// playTone(300, 120);
// delay(150);
if (pirState == LOW) {
// we have just turned on
Serial.println("Motion detected!");
// We only want to print on the output
change, not state
pirState = HIGH;
}
} else {
digitalWrite(ledPin, LOW); // turn LED OFF
playTone(0, 0);
delay(300);
if (pirState == HIGH){
// we have just turned off
Serial.println("Motion ended!");
// We only want to print on the output
change, not state
pirState = LOW;
}
}
// duration in mSecs, frequency in hertz
void playTone(long duration, int freq) {
duration *= 1000;
int period = (1.0 / freq) * 1000000;
long elapsed_time = 0;
while (elapsed_time < duration) {
digitalWrite(pinSpeaker, HIGH);
delayMicroseconds(period / 2);
digitalWrite(pinSpeaker, LOW);
delayMicroseconds(period / 2);
elapsed_time += (period);
}
}

```

- **OUTPUT** The output shows the LED and speaker turns ON when the sensor detects motion, and OFF when it doesn't.

B. Capacitance Measuring [12]

This measures the value of the capacitor using the arduino which then reads and write your result on the serial monitor.

- **COMPONENT**

- 1) Arduino board
- 2) 100pf Capacitor

- **CIRCUIT**

- **CODE [12]**

```

const int OUT_PIN = A2;
const int IN_PIN = A0;

//Capacitance between IN_PIN and Ground
//Stray capacitance is always present.

```

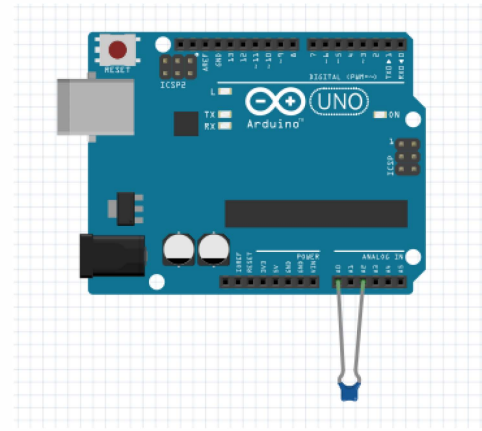


Fig. 4. Capacitance Measuring Circuit

```

Extra
capacitance can be added to
//allow higher capacitance to be
measured.
const float IN_STRAY_CAP_TO_GND = 24.48;
//initially this was 30.00
const float IN_EXTRA_CAP_TO_GND = 0.0;
const float IN_CAP_TO_GND
= IN_STRAY_CAP_TO_GND
+ IN_EXTRA_CAP_TO_GND;
const int MAX_ADC_VALUE = 1023;

void setup()
{
    pinMode(OUT_PIN, OUTPUT);
    //digitalWrite(OUT_PIN, LOW);
    //This is the default
    state for outputs
    pinMode(IN_PIN, INPUT);
    //digitalWrite(IN_PIN, LOW);

    Serial.begin(115200);
}

void loop()
{
    //Capacitor under test between
    OUT_PIN and IN_PIN
    //Rising high edge on OUT_PIN
    pinMode(IN_PIN, INPUT);
    digitalWrite(OUT_PIN, HIGH);
    int val = analogRead(IN_PIN);

    //Clear everything for next
    measurement
    digitalWrite(OUT_PIN, LOW);
    pinMode(IN_PIN, OUTPUT);

    //Calculate and print result

    float capacitance = (float)val *
    IN_CAP_TO_GND /
    (float)(MAX_ADC_VALUE - val);

```

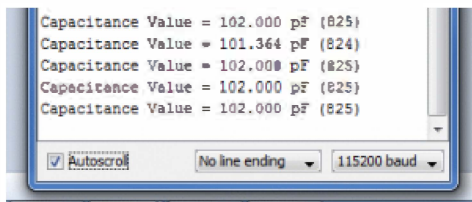


Fig. 5. Capacitance Measurin Output

```

Serial.print(F("Capacitance Value = "));
Serial.print(capacitance, 3);
Serial.print(F(" pF ("));
Serial.print(val);
Serial.println(F(") "));

while (millis() % 500 != 0)
    ;
}

```

- OUTPUT

IV. CONCLUSIONS

In this paper, We examined the working principle of an arduino uno both the hardware and software of the Arduino, from the components on the arduino Hardware to knowing how to write the code in the software (IDE), and how to combine both and construct your own project.

REFERENCES

- [1] P. D. Minns, *C Programming For the PC the MAC and the Arduino Microcontroller System*. Author House, 2013.
- [2] M. Banzi, *Getting Started with arduino*. " O'Reilly Media, Inc.", 2009.
- [3] Arduino, "Arduino uno." Last visited on 06/09/2014.
- [4] A. M. Gibb, *New media art, design, and the Arduino microcontroller: A malleable tool*. PhD thesis, Pratt Institute, 2010.
- [5] M. Margolis, *Arduino cookbook*. " O'Reilly Media, Inc.", 2011.
- [6] D. Mellis, M. Banzi, D. Cuartielles, and T. Igoe, "Arduino: An open electronic prototyping platform," in *Proc. CHI*, vol. 2007, 2007.
- [7] A. U. ARDUINO UNO, "Front. arduino uno board," 2012.
- [8] B. Stroustrup *et al.*, *The C++ programming language*. Pearson Education India, 1995.
- [9] J. Boxall, *Arduino Workshop: A Hands-on Introduction with 65 Projects*. No Starch Press, 2013.
- [10] "Intro to arduino," January 2014.
- [11] "Using motion detectors with an arduino," June 2012.
- [12] "Capacitance measurement with the arduino uno," January 2014.