

## Arduino Based Home AC Appliances Control Circuit Using IR-Signal

Aye Min<sup>1</sup> and Aye Yu Yu Naing<sup>2</sup>

### Abstract

Arduino based home AC appliances control circuit using IR-signal is constructed by using one AX1838 IR sensor, three C1383 transistors, three 12V relays, one Arduino UNO board and components available. The project aims in designing a system which is capable of controlling many electrical devices using the IR remote control. The IR data are transferred between IR remote and Arduino board. AX1838 IR-sensor is used to communicate between remote and Arduino. The circuit presented in this work is designed to control the three electrical devices such as TV, Fan, and electric bulb to turn on or turn off from a remote distance using control unit. The program for this project is written in C language and uploaded into the memory of the microcontroller on Arduino board by Arduino IDE window software. The operations of the whole circuit are controlled by software.

**Keywords:** Arduino UNO board, IR-sensor, Arduino IDE window software, C language

### Introduction

In this circuit, Arduino UNO board is used as a data receiver and it produces a digital signal (HIGH or LOW) to control the external devices. The Arduino can interact with almost any device that uses some form of remote control, including TVs, audio equipment, cameras, garage doors, appliances, and toys. Most remote controls work by sending digital data from a transmitter to a receiver using infrared light (IR) or wireless radio technology. An IR remote works by turning an LED on and off in patterns to produce unique codes. The codes are typically 12 to 32 bits (pieces of data). Each key on the remote is associated with a specific code that is transmitted when the key is pressed.

IR remote controls contain one or more special infrared light-generating LEDs that are used to transmit the IR signals. For example, when you press a button on the remote, the LED turns on and off repeatedly in a pattern that is unique for each button pressed on the remote. Arduino responds to IR remote signals using a device called an IR receiver module.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators. In simple terms, the Arduino is a tiny computer system that can be programmed with the instructions to interact with various forms of input and output. This project will demonstrate how to control digital output pins of Arduino using IR remote control. The block diagram of Arduino based home ac appliances control circuit using IR signal is shown in Figure (1).

<sup>1</sup>Lecturer, Department of Physics, Hinthada University

<sup>2</sup>M.Res. Student, Department of Physics, Hinthada University

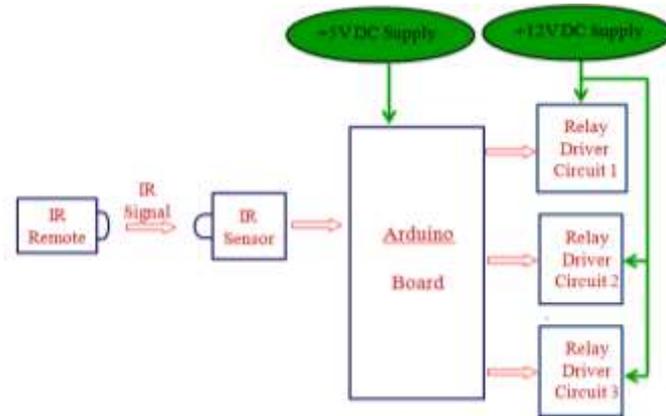


Figure (1) Block diagram of the system

### Arduino Hardware

The Arduino Uno has a second microcontroller onboard to handle all USB communication. The most important components of Arduino Uno board are shown in Figure (2).

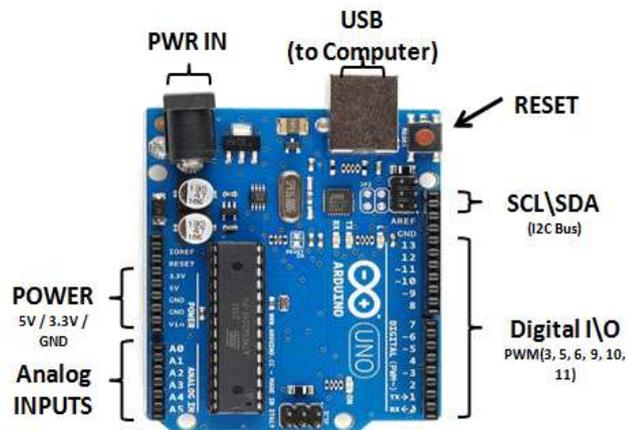


Figure (2) The Arduino’s most important components

On the far left of Figure (3) is the Universal Serial Bus (USB) connector. This connects the board to the computer for three reasons: to supply power to the board, to upload the instructions to the Arduino, and to send data to and receive it from a computer. On the right is the power connector.

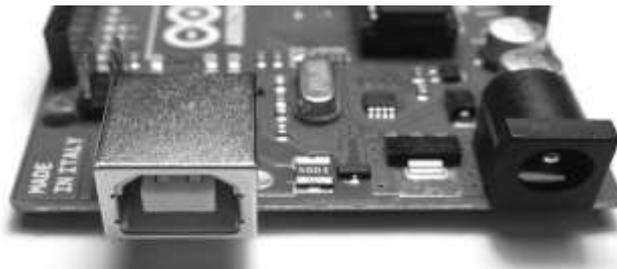


Figure (3) The USB and power connectors

The chip near the power connector is the heart of the board: the ATMEGA 328P-PU microcontroller, as shown in Figure (4). It is a tiny computer that contains a processor to

execute instructions, includes various types of memory to hold data and instructions from our sketches.

The two rows of small sockets are mounted at the left side of Arduino board, as shown in figure (5). The first row offers power connections and the ability to use an external RESET button. The second row offers six analog inputs that are used to measure electrical signals that vary in voltage.

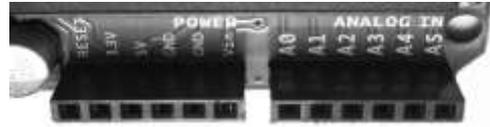
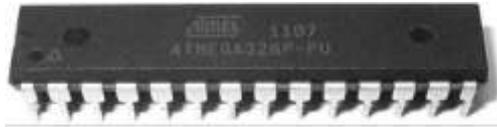


Figure (4) The ATMEGA328P-PU microcontroller Figure (5) The power and analog sockets

Sockets (or pins) numbered 0 to 13 on the right side of Arduino board are digital input/output (I/O) pins, as shown in Figure (6). They can either detect whether or not an electrical signal is present or generate a signal on command. Pins 0 and 1 are also known as the serial port, which is used to send and receive data to other devices. The pins labeled with a tilde (~) can also generate a varying electrical signal, which can be useful for such things as creating lighting effects or controlling electric motors. (Boxall, 2013)

The Arduino board has four LEDs: one on the far right labeled ON, which indicates when the board has power. The LEDs labeled *TX* and *RX* in Figure (7) light up when data is being transmitted or received between the Arduino and attached devices via the serial port and USB. The L LED is for your own use (it is connected to the digital I/O pin number 13).



Figure (6) The digital input/output pins

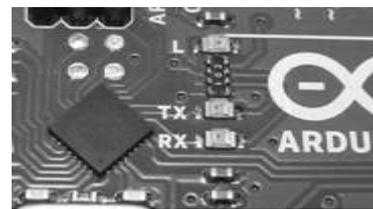


Figure (7) The onboard LEDs

### Getting Started With Arduino

The following steps need to start the Arduino project.

1. Download and install the Arduino development environment(IDE)
2. Connect the Arduino board to your computer via the USB cable
3. If needed, install the drivers
4. Launch the Arduino application and open the Blink example
5. Select your board and serial port
6. Upload your sketch

**Installing the Arduino IDE on Windows**

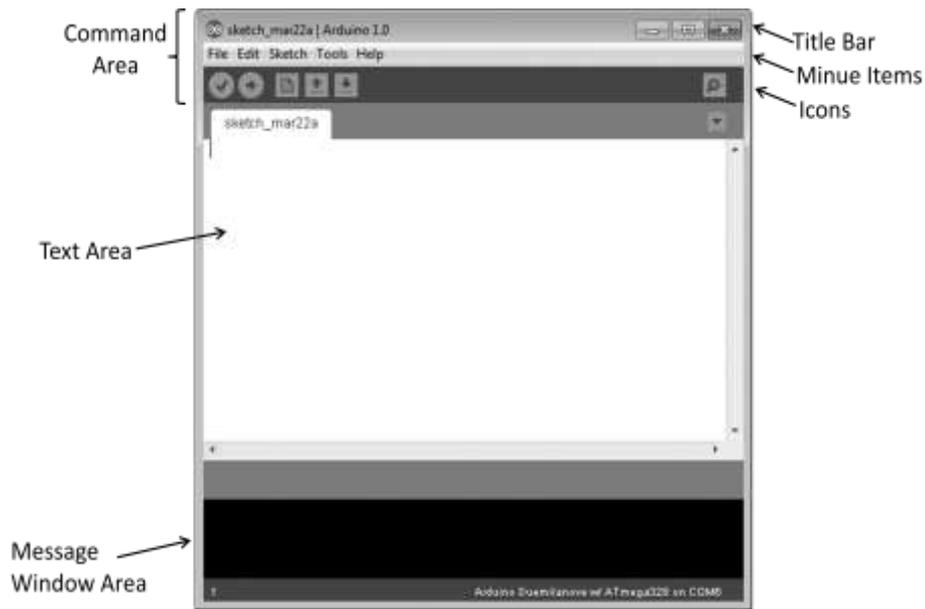


Figure (8) Arduino IDE main window

The Arduino IDE runs on all the latest versions of Microsoft Windows, such as Windows XP, Windows Vista, and Windows 7. The Arduino software for Windows, Mac, and Linux can be downloaded from <http://arduino.cc/en/Main/Software>. The IDE enables to write and edit code and convert this code into instructions that Arduino hardware understands. The IDE also transfers those instructions to the Arduino board (a process called uploading). The Arduino IDE software window is shown in Figure (8) (Craft, 2013).

**The IR Sensor (AX-1838)**



Figure (9) AX-1838 IR receiver

The independent IR receiver shown in Figure (9) is an AX-1838 IR sensor. The bottom leg of the receiver connects to an Arduino digital pin 11, the center leg to GND, and the top leg to 5V. The next step is to set up the IR receiver and test that it is working.

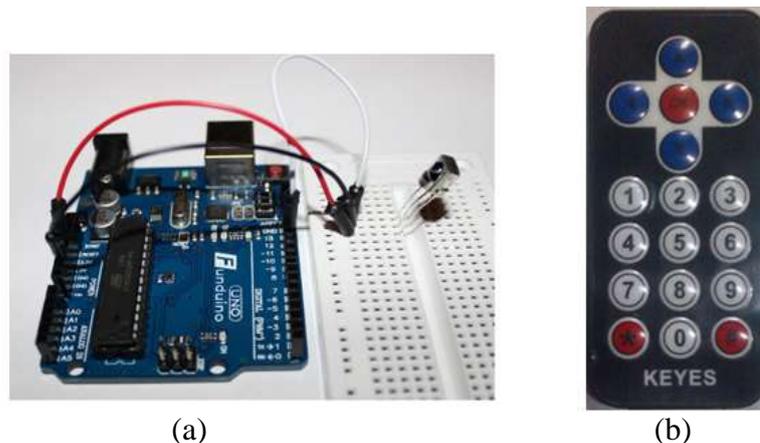


Figure (10) The connection of IR receiver to Arduino board and IR remote control

After connecting your IR receiver to the Arduino (see Figure 10-a), enter and upload the sketch to get the IR code for each button of IR-remote control (any company). The following is the sketch to get the IR code.

### The Sketch to Get IR Codes

```
#include <IRremote.h>

int receiverpin = 11;
IRrecv irrecv(receiverpin);
decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.print(results.value, HEX);
    Serial.print(" ");
    irrecv.resume( );
  }
}
```

When the sketch has uploaded to Arduino, open the Serial Monitor, aim the remote at the receiver, and start pressing buttons. After each button press, IR codes will display on the Serial Monitor. Note down the codes shown in the serial monitor for programming purpose. The control sketch (program) using these codes must be uploaded to Arduino board to control the three ac devices.

### The Control Sketch

```
#include <IRremote.h>
#include <IRremoteInt.h>

int RECV_PIN = 11;
int led1 = 2;
int led2 = 4;
int led3 = 7;
int itsONled[ ] = {0, 0, 0, 0};
#define code1 0xFF6897
#define code2 0xFF9867
```

```
#define code3 0xFFB04F
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}
void loop( )
{
  if (irrecv.decode(&results)){
    unsigned int value =results.value;
    switch(value) {
      case code1:
        if(itsONled[1] == 1)
        {
          digitalWrite(led1, LOW);
          itsONled[1] = 0;
        } else {
          digitalWrite(led1, HIGH);
          itsONled[1] = 1;
        }
        break;
      case code2:
        if(itsONled[2] == 1)
        {
          digitalWrite(led2, LOW);
          itsONled[2] = 0;
        } else {
          digitalWrite(led2, HIGH);
          itsONled[2] = 1;
        }
    }
```

```

break;
case code3:
  if(itsONled[3] == 1){
    digitalWrite(led3, LOW);
    itsONled[3] = 0;
  } else {
    digitalWrite(led3, HIGH);
    itsONled[3] = 1;
  }
  break;
}
Serial.println(value);
irrecv.resume();
}
}

```

### Circuit Construction

In this work, an Arduino Uno board was used as the heart of the control system. It was used to communicate the outside world such as IR-receiver and other peripheral devices. Three relay driver circuits were constructed by using three C1383 transistors and three 12V relays. They were connected to three digital I/O pins (2, 4, and 7) of Arduino to control the three electrical devices. The signal pin of AX-1838 IR-receiver was joined to Arduino digital pin 11, Vcc pin to 5V line, and GND pin to Arduino GND. 5V and 12V dc power supplies were constructed to operate the whole circuit. 5V DC is applied to Arduino board and 12V DC to three relay driver circuits. LM7805 and LM7812 regulator ICs are used in this supply circuit. The complete circuit of the whole system is shown in Figure (11) and circuit construction is shown in Figure (12).

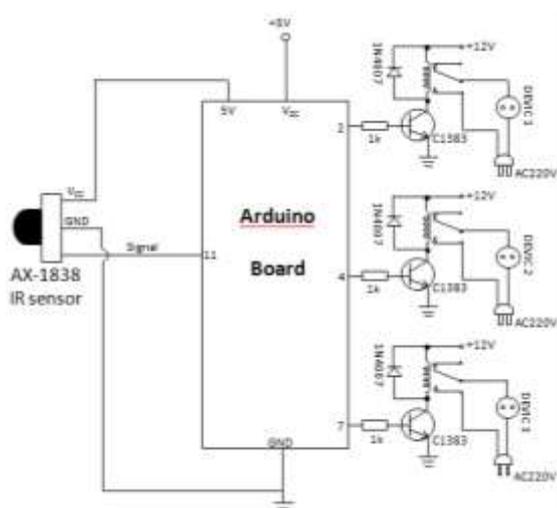


Figure (11) Complete circuit diagram



Figure (12) The construction of the whole circuit

### Operation of the Whole System

When the power is ON, take four digital I/O ports (2, 4, 7, and 11) of Arduino to low level setting. When a button on the remote is pressed, the LED turns on and off repeatedly in a pattern that is unique for each button pressed on the remote. This signal is received by a special IR receiver and converted to pulses of electrical current that are then converted to data in the receiver's electronics.

When the button '1' on remote control is pressed, the IR data of '1' is transmitted to receiver. Arduino receives this data (0xFF6897) via AX-1838 IR-sensor and produces the logic '1' (High) from digital I/O pin 2. It drives the transistor to saturate and the current flow through the relay coil. This is the 'ON' condition for device 1. When the button '1' is pressed again, Arduino receives the data again and produces logic '0' (LOW) from digital I/O pin 2. Then transistor is cutoff and device 1 is 'OFF'.

Similarly, to control the external devices 2 and 3, the button 2 and 3 on the remote must be pressed. The first press makes 'ON' the device and the second press 'OFF' it. Three relays RL1, RL2, and RL3 drive three electrical devices such as electric bulb, TV and fan.



Figure (13) The operation of the whole circuit

### Discussion and Conclusion

Arduino is best known for its hardware, but it also need software to program that hardware. Both the hardware and the software are called "Arduino." The combination enables to create projects that sense and control the physical world. The software is free, and open source. The boards are inexpensive and the hardware designs are also open source. The board can only control and respond to electricity, so specific components are attached to it to enable it to interact with the real world.

The IR remote control was used to control the electric appliances by Arduino Uno board. For example turning high power AC loads such as electric machine, motor, lights, fans, etc, ON or OFF. This system can control only three electric appliances. It can be further control several external devices by little modification in the sketch (program). If any invalid button (not defined in the software) on remote is pressed, Arduino board will not response for this action.

This circuit is very useful in field of electronic circuits. By using some modification it area of application can be extended in various fields. This circuit is generally used for a light, television, fan or similar electronic device that the person will want to turn on/off from bed.

This circuit is very low cost and is very useful to the elderly and physically challenged people. The system can be used in existing domestic area for either operating the loads through conventional switches or with the TV remote. But the IR-LED of the transmitter is required to be oriented towards the receiver when button on transmitter is pressed. This project gives us a great deal of knowledge about the Arduino board and C programming, working of the relay.

### **Acknowledgements**

We would like to express our profound thanks to Dr. Aung Win, Acting Rector and Dr. Theingi Shwe, Pro-Rector, Hinthada University, for their kind permission and encouragement to carry out this work. It is our pleasure to express our sincere thanks to Professor Dr. Than Tun, Head of Department of Physics, Hinthada University, for his kind permission and overall supporting during our research work. Special thanks are also due to Professor Dr. Win Win Thein, Department of Physics, Hinthada University, for her kind help and advice in preparing this work.

### **References**

Boxall, J., 2013. *Arduino Workshop*, No Starch Press: San Francisco.

Craft, B., 2013. *Arduino Projects For Dummies*, John Wiley & Sons, Ltd: Chichester, West Sussex, England

### **Online References**

<http://arduino.cc/en/Main/Software>

<http://github.com/shirriff/Arduino-IRremote/>

<http://www.efxkits.com>