EISYSTEM

Name:-Raghavendra balwan singh

College Name:- Bundelkhand University

ID NO:-18SESRDEL003

Contact No:-8840726315

Email:-raju229591@gmail.com

TRAINING REPORT ON EMBEDDED SYSTEM AND ROBOTICS

Submitted by:-

Raghavendra singh

Roll No:- 18SESRDEL003

Declaration

I Raghavendra singh ,student of Bundelkhand University Institute of Engineering Technology.5th Semester hereby declare that I have completed my vocational project in EISYSTEMS, Delhi in Academic year 2017-2018.the information is best and true as my knowledge.

Acknowledgement

I am truly thankful to all the faculties imparted lectures on various subject and topic and took us a plant in guided study visit along with detailed study of sensors and microcontroller. As a Electronics student mainly focused on a practical applications and concepts learn project like this one provided to me. I acknowledge the valuable input provided by MR.

Date:- 22 September 2018

Branch:-Electronics and Instrumentation

Semester:-5th

Training Report

Basic Electronics

Electronics is a branch of science that deals with the study of flow and controls electrons and study of their behaviour effects in vacuum, gases and semiconductor and devices with electrons. Control of electron accomplished by devices that resist, carry, select, steer, switch, store, manipulate, and exploit the electron.

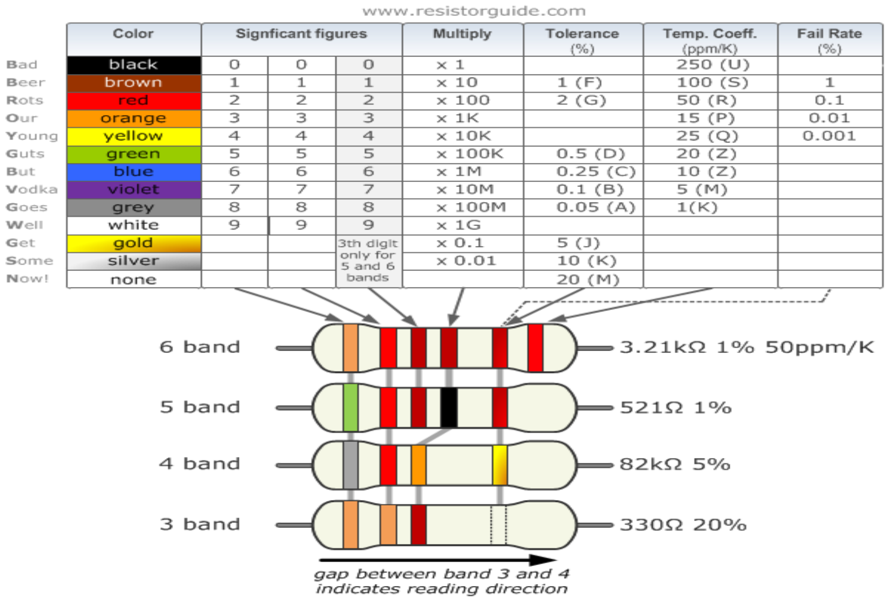
There are two types of element

1) ACTIVE:- Required source power to operate. Included transistor, Integrated circuits, LEDs, TRIACs, SCRs, etc

2) PASSIVE:- Operate without external power. Typical passive component are resistors, capacitor ,inductors and diodes.

Resistors

The resistance value for a resistor is generally indicated through the use of a 4 (or 3, or 5) band color code. For a typical four-band code, the first two bands tell you the first two digits of the resistance, the third band tells you how many zeros to add, and the fourth band tells you the tolerance. For a five-band color code, the first three bands tell you the first three digits of the resistance, the fourth band tells you how many zeros to add, and the fifth band tells you the tolerance



www.resistorguide.com

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

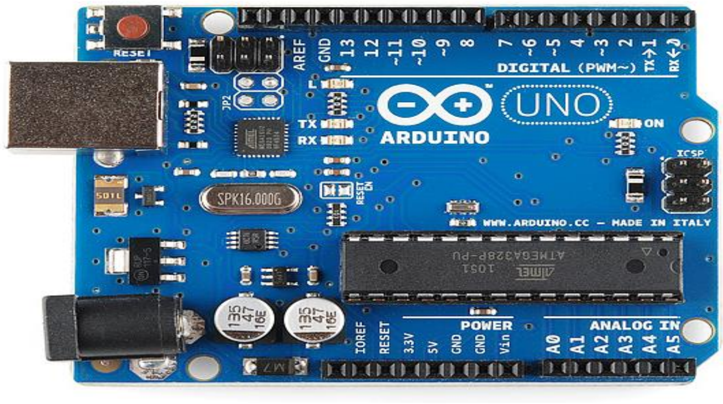
Embedded system

An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is designed for a specific function or for specific functions within a larger system. Industrial machines, agricultural and process industry devices, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys as well as mobile devices are all possible locations for an embedded system.

Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.



IR Sensor

An infrared sensor is an electronic instrument that is used to sense certain characteristics of its surroundings. It does this by either emitting or detecting infrared radiation. Infrared sensors are also capable of measuring the heat being emitted by an object and detecting motion.

Ultrasonic Sensor

This is a very simple and useful ultrasonic sensor. There are four pins that you would use to interface with the sensor: VCC, Trig (signal output pin), Echo (signal input pin), and GND. Each of the four pins are connected to the Arduino: VCC to 5v, Trig to a digital pin, Echo to a digital pin, and GND to GND (ground).

Jumper Wire

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Relay Module

comprises of an electromagnet and a contact unit. The definition is: Activating the contact unit using electromagnetic attraction, which is

produced when electric current exceeding the specified value flows to the electromagnet; the voltage and current (input signal) applied to the coil opens or shuts the contact.

WIFI Module

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another

application processor.This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the0front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP

applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.The applications of ESP8266 are Smart power plugs,Home automation ,Wi-Fi location-aware devices,Industrial wireless control,Security ID tags.

LCD (liquid crystal display)

LCD (liquid crystal display) is the technology used for displays in notebook and other smaller computers. Like light-emitting diode (LED) and

gas-plasma technologies, LCDs allow displays to be much thinner than cathode ray tube (CRT) technology. LCDs consume much less power than LED and gas-display displays because they work on the principle of blocking light rather than emitting it.

Arduino with C++

There isn't really an Arduino language as such. It's really just C++ with some domain-specific libraries. These add on various features, such as functions you can call to control the hardware. If you didn't have those functions, you'd need to fiddle directly with special registers to control everything. That's how embedded programming is usually done. It's fast, but it can be quite hard to learn and understand.

In addition to the functions, the libraries add alternative names for some types. For example, `boolean` and `byte` are not in the C++ standard. However, they are directly equivalent to `bool` and `unsigned char`.

All of these things mean you can probably port general C++ code directly to Arduino without difficulty. However, going back the other way may require some minor editing.

Project I made using simple arduino UNO

MOBILE CONTROL CAR

LINE FOLLOWER ROBOT

OBSTACAL AVOIDING CAR

NEVER FALLING CAR

TREE CLIMBING ROBOT

Internet Of Things

Arduino Cloud is an IoT application that helps makers build connected objects in a quick, easy and secure way. You can connect multiple devices to each other and allow them to exchange real-time data.

Programs

LFR(LINE FOLLEWER ROBOT)

```
/*------ Arduino Line Follower Code----- */
/*-------definning Inputs------*/
#define LS 2      // left sensor
#define RS 3      // right sensor
/*-------definning Outputs------*/
#define LM1 4       // left motor
#define LM2 5       // left motor
#define RM1 6       // right motor
#define RM2 7       // right motor
void setup()
{
  pinMode(LS, INPUT);
  pinMode(RS, INPUT);
  pinMode(LM1, OUTPUT);
  pinMode(LM2, OUTPUT);
  pinMode(RM1, OUTPUT);
  pinMode(RM2, OUTPUT);
}
void loop()
{
  if(digitalRead(LS) && digitalRead(RS))     // Move Forward
  {
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, LOW);
  }

  if(!(digitalRead(LS)) && digitalRead(RS))     // Turn right
  {
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, LOW);
  }

  if(digitalRead(LS) && !(digitalRead(RS)))     // turn left
  {
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
  }

  if(!(digitalRead(LS)) && !(digitalRead(RS)))     // stop
  {
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
  }
}
```

Mobile Controlled ROBOT

```
#define m11 11    // rear motor
#define m12 12
#define m21 10    // front motor
#define m22 9
char str[2],i;
void forward()
{
   digitalWrite(m11, LOW);
   digitalWrite(m12, LOW);
   digitalWrite(m21, HIGH);
   digitalWrite(m22, LOW);
}
void backward()
{
   digitalWrite(m11, LOW);
   digitalWrite(m12, LOW);
   digitalWrite(m21, LOW);
   digitalWrite(m22, HIGH);
}
void left()
{
   digitalWrite(m11, HIGH);
   digitalWrite(m12, LOW);
   delay(100);
   digitalWrite(m21, HIGH);
   digitalWrite(m22, LOW);
}
void right()
{
   digitalWrite(m11, LOW);
   digitalWrite(m12, HIGH);
   delay(100);
   digitalWrite(m21, HIGH);
   digitalWrite(m22, LOW);
}
void Stop()
{
   digitalWrite(m11, LOW);
   digitalWrite(m12, LOW);
   digitalWrite(m21, LOW);
   digitalWrite(m22, LOW);
}
void setup()
{
  Serial.begin(9600);
  pinMode(m11, OUTPUT);
  pinMode(m12, OUTPUT);
  pinMode(m21, OUTPUT);
  pinMode(m22, OUTPUT);
}
```

```
void loop()
{
  while(Serial.available())
  {
    char ch=Serial.read();
    str[i++]=ch;

    if(str[i-1]=='1')
    {
     Serial.println("Forward");
     forward();
     i=0;
    }
    else if(str[i-1]=='2')
    {
     Serial.println("Left");
     right();
     i=0;
    }
    else if(str[i-1]=='3')
    {
      Serial.println("Right");
      left();
      i=0;
    }

    else if(str[i-1]=='4')
    {
      Serial.println("Backward");
      backward();
      i=0;
    }
    else if(str[i-1]=='5')
    {
      Serial.println("Stop");
      Stop();
      i=0;
    }
    delay(100);
  }
}
```