Software Used:

Arduino IDE

<https://www.arduino.cc/en/software>

Components Required:

* 1\*ESP32 cam module
* 2\*Arduino Uno
* 1\*IR sensor array
* 4\*N20 motor and wheels
* L298n Motor driver
* 2\* 18650 li-ion battery and battery holder
* Jumper Cables

Pin Connections

Arduino 1 with IR sensor array

IR sensor-->Arduino

OUT5-->2

OUT4-->3

OUT3-->4

OUT2-->5

OUT1-->6

VCC-->5V

GND-->GND

Arduino 1 with motor driver

Motor Driver-->Arduino 1

MA1-->7

MA2-->8

MB1-->12

MB2-->13

ENA-->10

ENB-->11

12V-->Battery +ve

5V-->Arduino 5V

GND-->Battery -ve + Arduino GND

Arduino 2 with ESP32 Cam module

ESP32-->Arduino 2

U0T-->TX

U0R-->RX

5V--> Arduino 5V

GND-->Arduino GND

GPO0 ESP32--> GND ESP32 (It has to be pulled out after uploading the code and then the Reset button is to be pressed to get the Output IP address on the serial monitor)

CODE

1-Code for interfacing of IR sensor array , Motor Driver and Arduino Uno

#define m1 7 //Right Motor MA1

#define m2 8 //Right Motor MA2

#define m3 12 //Left Motor MB1

#define m4 13 //Left Motor MB2

#define e1 10 //Right Motor Enable Pin EA

#define e2 11 //Left Motor Enable Pin EB

//5 Channel IR Sensor Connection//

#define ir1 2

#define ir2 3

#define ir3 4

#define ir4 5

#define ir5 6

void setup() {

Serial.begin(9600);

pinMode(m1, OUTPUT);

pinMode(m2, OUTPUT);

pinMode(m3, OUTPUT);

pinMode(m4, OUTPUT);

pinMode(e1, OUTPUT);

pinMode(e2, OUTPUT);

pinMode(ir1, INPUT);

pinMode(ir2, INPUT);

pinMode(ir3, INPUT);

pinMode(ir4, INPUT);

pinMode(ir5, INPUT);

}

void loop() {

//Reading Sensor Values

int s1 = digitalRead(ir1); //Left Most Sensor

int s2 = digitalRead(ir2); //Left Sensor

int s3 = digitalRead(ir3); //Middle Sensor

int s4 = digitalRead(ir4); //Right Sensor

int s5 = digitalRead(ir5); //Right Most Sensor

//if only middle sensor detects black line

if ((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 1))

{

//going forward with full speed

analogWrite(e1, 80);

analogWrite(e2, 80);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if only left sensor detects black line

if ((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 1) && (s5 == 1))

{

//going left with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//if only left most sensor detects black line

if ((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))

{

//going left with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//if only right sensor detects black line

if ((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 0) && (s5 == 1))

{

//going right with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if only right most sensor detects black line

if ((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

{

//going right with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if middle and right sensor detects black line

if ((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 1))

{

//going left with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if middle and left sensor detects black line

if ((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))

{

//going right with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//if middle, left and left most sensor detects black line

if ((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))

{

//going right with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//if middle, right and right most sensor detects black line

if ((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 0))

{

//going left with full speed

analogWrite(e1, 75);

analogWrite(e2, 75);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if all sensors are on a black line

if ((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))

{

//stop

digitalWrite(m1, LOW);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, LOW);

}

}

1. Code for interfacing of ESP32 cam module and Arduino Uno

#include "esp\_camera.h"

#include <WiFi.h>

#define CAMERA\_MODEL\_AI\_THINKER // Has PSRAM

#include "camera\_pins.h"

// WiFi credentials

const char\* ssid = "xYzee";

const char\* password = "mainak1234";

void startCameraServer();

void setup() {

Serial.begin(115200);

Serial.setDebugOutput(true);

Serial.println();

camera\_config\_t config;

config.ledc\_channel = LEDC\_CHANNEL\_0;

config.ledc\_timer = LEDC\_TIMER\_0;

config.pin\_d0 = Y2\_GPIO\_NUM;

config.pin\_d1 = Y3\_GPIO\_NUM;

config.pin\_d2 = Y4\_GPIO\_NUM;

config.pin\_d3 = Y5\_GPIO\_NUM;

config.pin\_d4 = Y6\_GPIO\_NUM;

config.pin\_d5 = Y7\_GPIO\_NUM;

config.pin\_d6 = Y8\_GPIO\_NUM;

config.pin\_d7 = Y9\_GPIO\_NUM;

config.pin\_xclk = XCLK\_GPIO\_NUM;

config.pin\_pclk = PCLK\_GPIO\_NUM;

config.pin\_vsync = VSYNC\_GPIO\_NUM;

config.pin\_href = HREF\_GPIO\_NUM;

config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;

config.pin\_sscb\_scl = SIOC\_GPIO\_NUM;

config.pin\_pwdn = PWDN\_GPIO\_NUM;

config.pin\_reset = RESET\_GPIO\_NUM;

config.xclk\_freq\_hz = 20000000;

config.frame\_size = FRAMESIZE\_UXGA;

config.pixel\_format = PIXFORMAT\_JPEG; // for streaming

//config.pixel\_format = PIXFORMAT\_RGB565; // for face detection/recognition

config.grab\_mode = CAMERA\_GRAB\_WHEN\_EMPTY;

config.fb\_location = CAMERA\_FB\_IN\_PSRAM;

config.jpeg\_quality = 12;

config.fb\_count = 1;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality

// for larger pre-allocated frame buffer.

if(config.pixel\_format == PIXFORMAT\_JPEG){

if(psramFound()){

config.jpeg\_quality = 10;

config.fb\_count = 2;

config.grab\_mode = CAMERA\_GRAB\_LATEST;

} else {

// Limit the frame size when PSRAM is not available

config.frame\_size = FRAMESIZE\_SVGA;

config.fb\_location = CAMERA\_FB\_IN\_DRAM;

}

} else {

// Best option for face detection/recognition

config.frame\_size = FRAMESIZE\_240X240;

#if CONFIG\_IDF\_TARGET\_ESP32S3

config.fb\_count = 2;

#endif

}

#if defined(CAMERA\_MODEL\_ESP\_EYE)

pinMode(13, INPUT\_PULLUP);

pinMode(14, INPUT\_PULLUP);

#endif

// camera init

esp\_err\_t err = esp\_camera\_init(&config);

if (err != ESP\_OK) {

Serial.printf("Camera init failed with error 0x%x", err);

return;

}

sensor\_t \* s = esp\_camera\_sensor\_get();

// initial sensors are flipped vertically and colors are a bit saturated

if (s->id.PID == OV3660\_PID) {

s->set\_vflip(s, 1); // flip it back

s->set\_brightness(s, 1); // up the brightness just a bit

s->set\_saturation(s, -2); // lower the saturation

}

// drop down frame size for higher initial frame rate

if(config.pixel\_format == PIXFORMAT\_JPEG){

s->set\_framesize(s, FRAMESIZE\_QVGA);

}

#if defined(CAMERA\_MODEL\_M5STACK\_WIDE) || defined(CAMERA\_MODEL\_M5STACK\_ESP32CAM)

s->set\_vflip(s, 1);

s->set\_hmirror(s, 1);

#endif

#if defined(CAMERA\_MODEL\_ESP32S3\_EYE)

s->set\_vflip(s, 1);

#endif

WiFi.begin(ssid, password);

WiFi.setSleep(false);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");

Serial.print(WiFi.localIP());

Serial.println("' to connect");

}

void loop() {

delay(10000);

}

For reference (ESP32 CAM MODULE )

STEP1-Menu Arduino IDE > Preferences > Settings > Additional board manager URLs. Add “https://dl.espressif.com/dl/package\_esp32\_index.json”

STEP2-File -> Examples -> ESP32 -> Camera -> CameraWebServer.

STEP3-Now open Tools -> Board-> ESP32 Arduino -> select : ESP32 Wrover module

Set the Tools menu settings as shown below

Board: “ESP32 Wrover Module”

Upload Speed: “115200”

Flash Frequency: “80MHz”

Flash Mode: “Q10”

Partition Scheme: “Huge APP (3MB No OTA/1 MB SPIFFS)”

Core Debug Level: “None”

Port: “COMx’ > According to your port connection

<https://www.circuitschools.com/how-to-program-upload-the-code-to-esp32-cam-using-arduino-or-programmer/>