LAB: Smart mini-fan with STM32-duino

Date : 2023.09.09

Author/Partner: Duwon Yang / Yoonseok Choi

Github : LINK

Demo Video : LINK

I. Introduction

In this lab, We create a simple program about an automatic mini-fan that runs only when the face is near the fan. Additionally, it has 3steps for the fan velocity(0%, 50%, 100%). We uses arduino IDE for implementing this program.

Hardware

- MCU
  - NUCLEO-F401RE
- Sensor:
  - Ultrasonic distance sensor(HC-SR04) x1
- Actuator/Display
  - DC motor(RK-280RA) x1
- LED

Software

- Arduino IDE

II. Procedure

The program needs to run the Fan only when the distance of an object is within a certain value.

- As the button B1 is pressed, change the fan velocity. the MODE(states) are
  - MODE(state): OFF(0%), MID(50%), HIGH(100%)
- When the object(face) is detected about 50mm away, then it automatically pauses the fan temporarily.
  - Even the fan is temporarily paused, the MODE should be changed whenever the button B1 is pressed
- When the object(face) is detected within 50mm, then it automatically runs the fan
  - It must run at the speed of the current MODE
- LED(LED1): Turned OFF when MODE OFF. Otherwise, blink the LED with 1 sec period(1s ON, 1s OFF)
- Print the distance and PWM duty ratio in Tera-Term console (every 1 sec).
- Must use Mealy FSM to control the mini-fan
  - Draw a FSM(finite-state-machine) table and state diagram

III. Configuration

Ultrasonic distance sensor

Trigger:

- Generate a trigger pulse as PWM to the sensor
- Pin : D10 (TIM4 CH1)
- PWM out: 50ms period, 10us pulse-width

Echo:

- Receive echo pulses from the ultrasonic sensor
- Pin : D7 (Timer1 CH1)
- Measure the distance by calculating pulse-width of the echo pulse.

USART

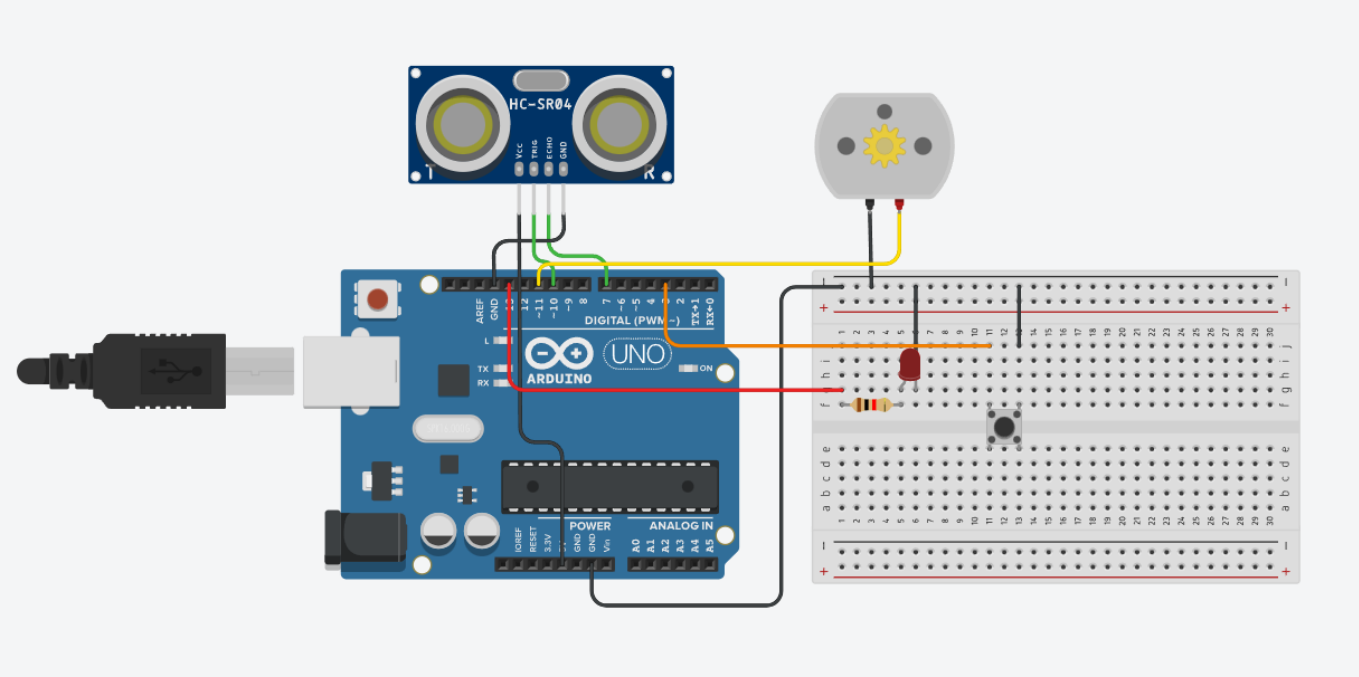
- Display measured distance in [cm] on serial monitor of Tera-Term.
- Baudrate 9600

DC Motor

- PWM: PWM1, set 10ms of period by default
- Pin: D11 (Timer1 CH1N)

Circuit/Wiring Diagram

External circuit diagram that connects MCU pins to peripherals(sensor/actuator)



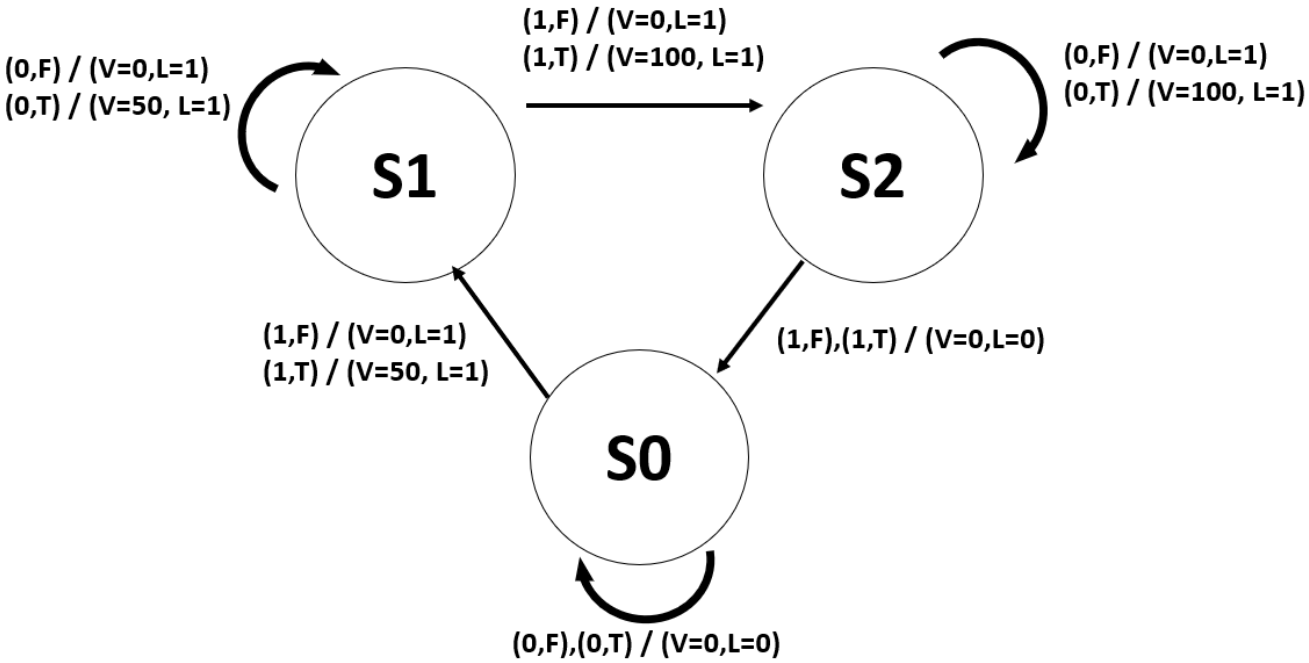
IV. Algorithm

Overview

Mealy FSM Table

Present State	Next State (X, Y)				Output Z			
	(0, F)	(0, T)	(1, F)	(1, T)	(0, F)	(0, T)	(1, F)	(1, T)
S0	S0	S0	S1	S1	V=0 L=0	V=0 L=0	V=0 L=1	V=50 L=1
S1	S1	S1	S2	S2	V=0 L=1	V=50 L=1	V=0 L=1	V=100 L=1
S2	S2	S2	S0	S0	V=0 L=1	V=100 L=1	V=0 L=0	V=0 L=0

Mealy State Diagram



Description with Code

- Lab source code: LINK
- Description1 Define State/Output Form/Function and Pin, and declare multiple variables.

```
#define S0  0
#define S1  1
#define S2  2
#define OFF LOW
#define ON HIGH
```

```
void nextState();
void pressed();
void stateOutput();
//핀 설정
const int trigPin = 10;
const int echoPin = 7;
const int ledPin = 13;
const int pwmPin = 11;
const int btnPin = 3;

long interval = 1000;
long curtime;
long pretime;
long duration;
long distance;
int state = S0;
int ledState = 0;
int pwmState = 0;
int pwmOut = 0;
int ledOut = LOW;
int input[2] = {0, 0};
```

- Description2

Define State table components and set arrays according to above FSM table.

```
typedef struct {
  unsigned int next[2][2]; // nextstate = FSM[state].next[inputX][inputY] X에 의해
서만 구분가능
  unsigned int vel[2][2]; // velocity = FSM[state].vel[inputX][inputY]
  unsigned int led[2][2]; // led = FSM[state].led[inputX][inputY]
} State_t;

State_t FSM[3] = {
{{{S0,S0}, {S1,S1}},{{0,0},{0,50}},{{OFF,OFF},{ON,ON}}},
{{{S1, S1},{S2, S2}},{{0,50},{0,100}},{{ON,ON},{ON,ON}}},
{{{S2, S2},{S0, S0}},{{0,100},{0,0}},{{ON,ON},{OFF,OFF}}},
};
```

- Description3

setup led pin, button pin, motor pin, and Ultrasonic distance pin. additionally add interrupt function (when pressed button, do pressed).

```
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
  pinMode(pwmPin, OUTPUT);

  pinMode(btnPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(btnPin), pressed, FALLING);
}
```

- Description4

Generate pwm signal on the trigger pin and calculate distance using how much time it takes.

update State and output. then, write pwmOut, ledOut value to pwm, led.

finally, print input value[1] (whether face nearby), state, distance.

```
void loop() {
  //Generate pwm signal on the trigger pin.
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(10);

  //Distance is calculated using how much time it takes.
  duration = pulseIn(echoPin, HIGH);
  distance = (float)duration / 58.0;


  //Calculate next state. then update State
  nextState();
  // Output State
  stateOutput();

  analogWrite(pwmPin, pwmOut); //pwm에 입력
  digitalWrite(ledPin, ledOut); //ledState에 입력



  Serial.print("input : ");
  Serial.print(input[1]);
  Serial.print(", state : ");
  Serial.print(state);

  Serial.print(", distance = ");
  Serial.print(distance);
  Serial.println(" [cm]");
  delay(200);
}
```

- Description5 when the button pressed, change input[0] = 1, then apply to nextState function.

wrap up by initializing to input[0] = 0

```
void pressed(){
  input[0] = 1;
  nextState();
  input[0] = 0;
}
```

- Description6

when the distance is less than 10cm, update input[1] = 1

if not, update input[1] = 0

Finish and update the state

```
void nextState(){
  if(distance < 10)
    input[1] = 1;
  else
    input[1] = 0;
  state = FSM[state].next[input[0]][input[1]];
}
```

- Description7

receive current time data at variable [curtime] from function millis().

if state equals 0, ledOut is updated by FSM.

if not, calculate interval time[curtime-pretime] and compare with variable [interval].

at there, if interval time[curtime-pretime] is more than [interval], update curtime to pretime, toggle ledOut, and output led.

```
void stateOutput(){
  curtime = millis();
  pwmOut = FSM[state].vel[input[0]][input[1]];
  if(state == 0)
    ledOut = FSM[state].led[input[0]][input[1]];
  else{
    if(curtime-pretime>=interval){
      pretime = curtime;
      ledOut ^= 1;
      digitalWrite(ledPin, ledOut);
    }
  }
}
```

Results and Analysis

Results

At this experiment, We implemented a smart mini fan that has 3 output modes(0, 50, 100) and operates only when approached under 10cm. even more, the LED is designed to turn off or on at 1 second intervals if it is in a state where it can work when it is close under 10cm.

Demo Video

LINK

Analysis

According to 3 output modes, system operates only approached under 10cm. in each state, DC motor has specific velocity (0, 50, 100). the LED is designed to turn off or on at 1 second intervals if it is in a state where it can work when it is close under 10cm.

Reference

https://ykkim.gitbook.io/ec/ec-course/lab/lab-report-template