LAB : Smart mini -fan with STM 32- duino

Embedded Controller Lab Report

LAB: Smart mini-fan with STM32-duino

Date : 2023.09.15

Author/Partner: 21900213 김한민, 21900253 문성빈

Github: https://github.com/ansterdaz/HKim/blob/main/LAB%201

Demo Video: https://youtu.be/41KSXC9E660?si=14hAmMWPbfr_NZU

I. Introduction

This lab operates a smart minifan using a DC motor, detection sensor, and STM32. The purpose of this lab is to design the logic for fan operation and implement the operation using Arduino IDE.

Hardware

MCU

- NUCLEO-F401RE

Sensor/Actuator:

- Ultrasonic distance sensor(HC-SR04) x1
- DC motor (RK-280RA)

Software

- Arduino IDE

II. Procedure

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

Example: An automatic mini-fan that runs only when the face is near the fan. Otherwise turns off.

- 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 50 mm 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.
- 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

=> Example Table. See below for example code.

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)
- Input Capture: Input mode
- Measure the distance by calculating pulse-width of the echo pulse.

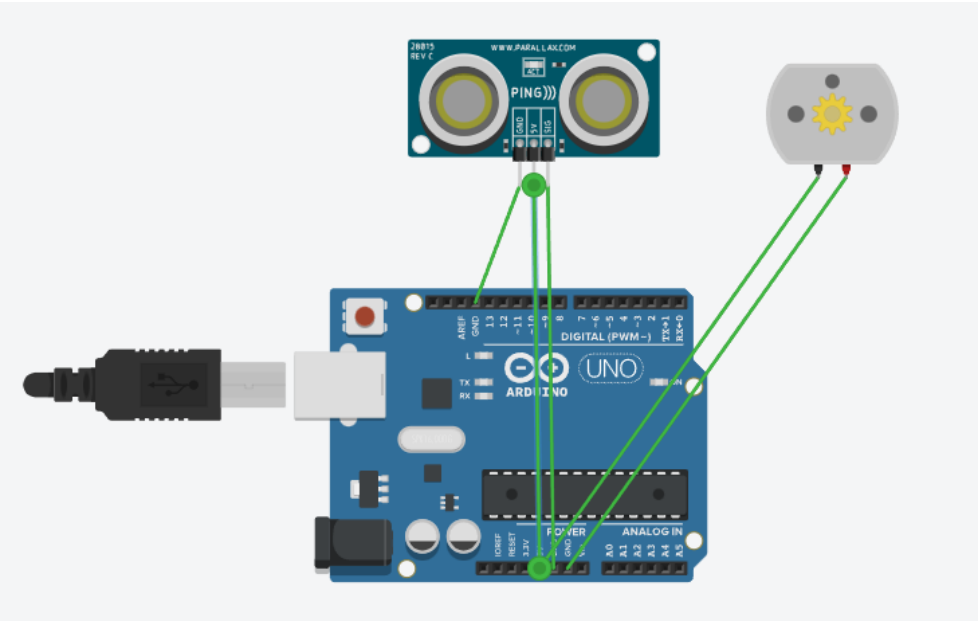
USART

- 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)

IV. Circuit/Wiring Diagram



V. Algorithm

Overview

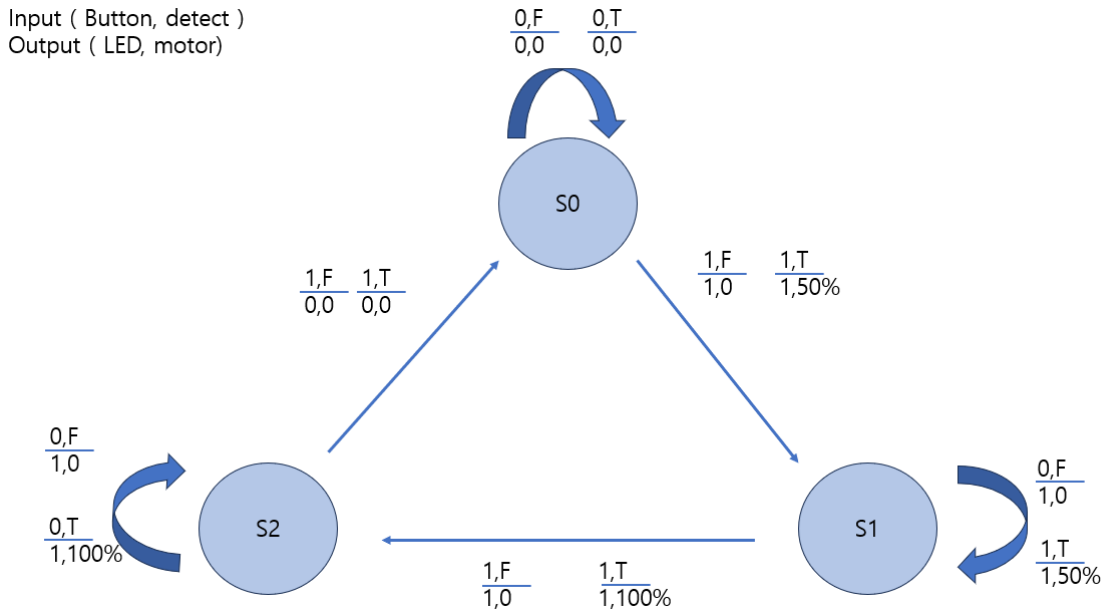
input 1 LED = 0 , 1

input 2 Sensor = F , T

Output 1 motor = 0 ,50,100

Output 2 LED = HIGH , LOW

Mealy FSM Diagram



Mealy FSM Table

Present State	Next State				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 = LOW	V = 0 L = LOW	V = 0 L = HIGH	V = 50 L = HIGH
S1	S1	S1	S2	S2	V = 0 L = HIGH	V = 0 L = HIGH	V = 0 L = HIGH	V = 100 L = HIGH
S2	S1	S2	S0	S0	V = 0 L = HIGH	V = 100 L = HIGH	V = 0 L = LOW	V = 0 L = LOW

VI. Description with Code

- Lab source code: https://github.com/ansterdaz/HKim/blob/main/LAB%201

- Description 1

```
typedef struct {

unsigned int motor[2][2];      // output    = FSM[state].motor out[input X][input Y]
unsigned int led[2][2];        // output    = FSM[state].led out[input X][input Y]
unsigned int next[2][2];       // nextstate = FSM[state].next[input X][input Y]

} State_t;
```

=> A structure was created according to the created state table. At this time, slightly differently from the table, the output was divided into motor and LED.

- Description 2

```
State_t FSM[3] = {

{ {{0,0},  {0,50} },     {{LOW,HIGH},  {HIGH,HIGH}}  ,  { {S0,S0},{S1,S1}}  },
{ {{0,50}, {0,100}},     {{LOW,HIGH},  {HIGH,HIGH}}  ,  { {S1,S1},{S2,S2}}  },
{ {{0,100},{0,0}  },     {{HIGH,HIGH}, {LOW,LOW}  }  ,  { {S2,S2},{S0,S0}}  }

};
```

=> The code was written according to the state table.

- Description 3

```
void stateOutput(){
pwmOut = FSM[state].motor[input[0]][input[1]];
ledOut = FSM[state].led[input[0]][input[1]];

}
```

=> Code was written to define the output of motor and LED.

VII. Results and Analysis

Result

1. When the button is pressed, the LED turns on, and when the sensor is detected, the motor speed operates at 50, and when not detected, the motor speed is 0.
2. When the button is pressed, the LED turns on, when the sensor detects, the motor speed operates at 100, and when not detected, the motor speed is 0.
3. When the button is pressed the button, the LED turns off and the motor speed is 0.

Analysis

1. When the button is pressed, the state is changed from S0 to S1 and the LED is turned on. The motor operates at a speed of 50 when the sensor is detected. The motor does not work when it is not detected.
2. When the button is pressed, the state is changed from S1 to S2, the LED is still on, and the sensor is detected, and the motor operates at a speed of 100. The motor does not work when it is not detected.
3. When the button is pressed, the state changes from S2 to S1, the LED turns off, and the motor does not work.

Demo Video

Link: https://youtu.be/41KSXC9E660?si=14hAmMWPbfr__NZU

VIII. Reference

https://ykkim.gitbook.io/ec/ec-course/lab/lab-smart-mini-fan-with-stm32-duino