VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

CC02 – Lab Report

MICROCONTROLER

Supervisors: Nguyen Thien An
Students: Vu Trinh Thanh Binh 2252085

Ho Chi Minh City, December 10, 2024

Contents

1 Exercise

The GitHub link for the lab schematics is at here or in this link: https://github.com/thanhbinh0710/VXL.git.

1.1 Proteus Schematic

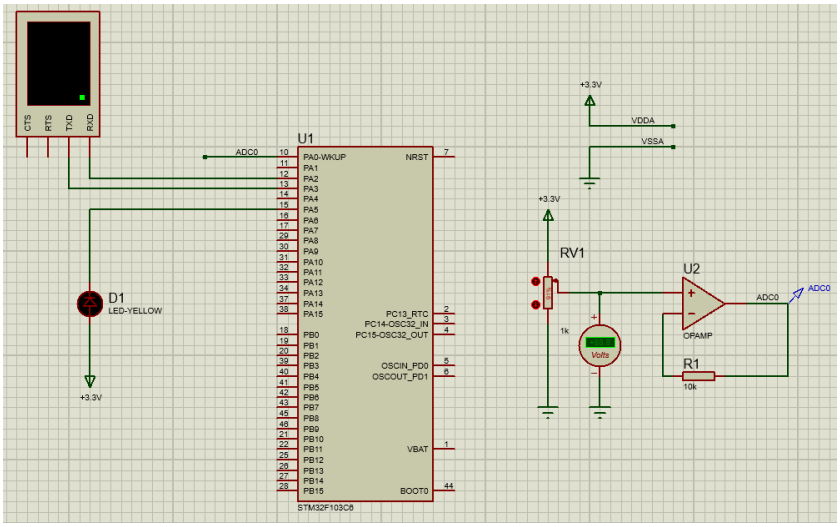

Figure 1: *Schematic*

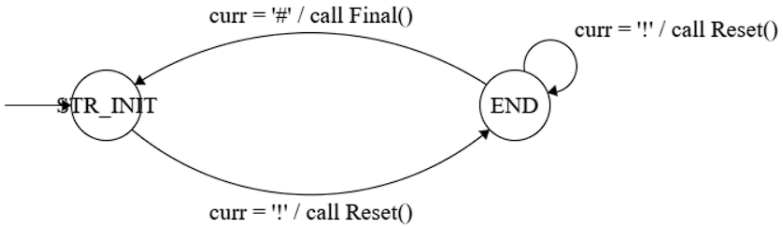

Figure 2: *Parser fsm*

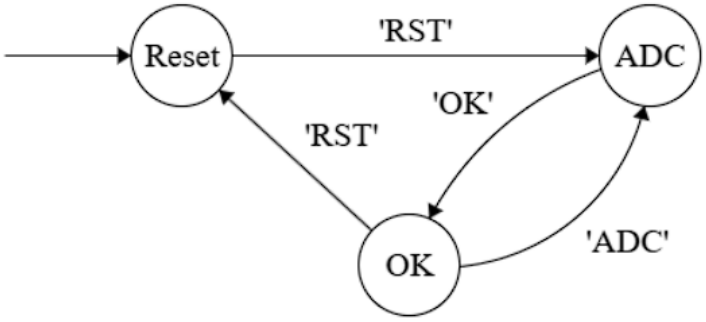

Figure 3: *UART fsm*

1.2 Parser Command Source Code:

```
#include "parser.h"

void reset(){
  commandIndex = 0;
}

void add(char c){
  if (commandIndex < MAX_BUFFER - 1){
    command[commandIndex++] = c;
  }
}

void final(){
  command[commandIndex] = '\0';
  commandFlag = 1;
}

void parser_fsm(){
  char curr = temp;
  switch (Pstatus){
  case STR_INIT:
    if (curr == '!'){
      Pstatus = END;
      reset();
    }
    break;
  case END:
    if (curr == '#'){
      final();
      Pstatus = STR_INIT;
    }else if (curr == '!'){
      reset();
    }else{
      add(curr);
    }
    break;
  default:
    break;
  }
}
```

1.3 UART Source Code:

- Handle Resest Function:

```
void HandleReset(ADC_HandleTypeDef  hadc1, UART_HandleTypeDef huart2){
  if (commandFlag == 1){
    commandFlag = 0;
    if (command[0] == 'R' && command[1] == 'S' && command[2] == 'T'){
      HAL_ADC_Start(&hadc1);
      ADCvalue = HAL_ADC_GetValue(&hadc1);
      HAL_ADC_Stop(&hadc1);

      HAL_UART_Transmit(&huart2,(uint8_t*)str, sprintf(str, "\r\n"), 1000);
      Ustatus = ADC;
      setTimer(1, 3000);

    }
  }
}
```

- Handle Getting ADCvalue:

```
    void HandleADC(UART_HandleTypeDef huart2){
  HAL_UART_Transmit(&huart2, (uint8_t*)str,sprintf(str,"!ADC=%ld\r\n", ADCvalue),
    1000);
  Ustatus = OK;
}
```

- Handle ”OK” Function:

```
    void HandleOK(UART_HandleTypeDef huart2){
  if (commandFlag == 1){
    commandFlag = 0;
    if (command[0] == 'O' && command[1] == 'K'){
      HAL_UART_Transmit(&huart2, (uint8_t*)str, sprintf(str,"\r\n"), 1000);
      Ustatus = Reset;
      clearTimer(1);
    }
  }
  if (timer_flag[1] == 1){
    Ustatus = ADC;
    setTimer(1,3000);
  }
}
```

- UART Structure:

```
    void UART_fsm(ADC_HandleTypeDef hadc1, UART_HandleTypeDef huart2){
  switch (Ustatus){
  case Reset:
    HandleReset(hadc1, huart2);
    break;
```

```
  case ADC:
    HandleADC(huart2);
    break;
  case OK:
    HandleOK(huart2);
    break;
  default:
    break;
  }
}
```

1.4 Callback Function:

```
    void HAL_UART_RxCpltCallback ( UART_HandleTypeDef * huart ) {
  //Luu ki tu nhan dc vao buffer
  if (huart->Instance == USART2){
    buffer[bufferIndex++] = temp;
    if (bufferIndex == MAX_BUFFER) bufferIndex = 0;
    bufferFlag = 1;
    //Truyen ki tu nhan dc qua UART
    HAL_UART_Transmit (&huart2 , &temp , 1 , 50) ;
    //Nhan ki tu moi
    HAL_UART_Receive_IT (&huart2 , &temp , 1) ;
  }
}
```

- Main loop:

```
    setTimer(0,1000);
  clearTimer(1);
  while (1)
  {
    /* USER CODE END WHILE */
    if (timer_flag[0] == 1){
      HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
      setTimer(0,1000);
    }
    if (bufferFlag == 1){ //Neu co ki tu moi trong buffer
      parser_fsm(); //Goi ham nay de xu ly
      bufferFlag = 0;
    }
    UART_fsm(hadc1, huart2);
    /* USER CODE BEGIN 3 */
  }
```

References