Ho Chi Minh city University of Technology
Computer Science and Engineering Faculty

Project – LAB5

Microcontroller – Microprocessor

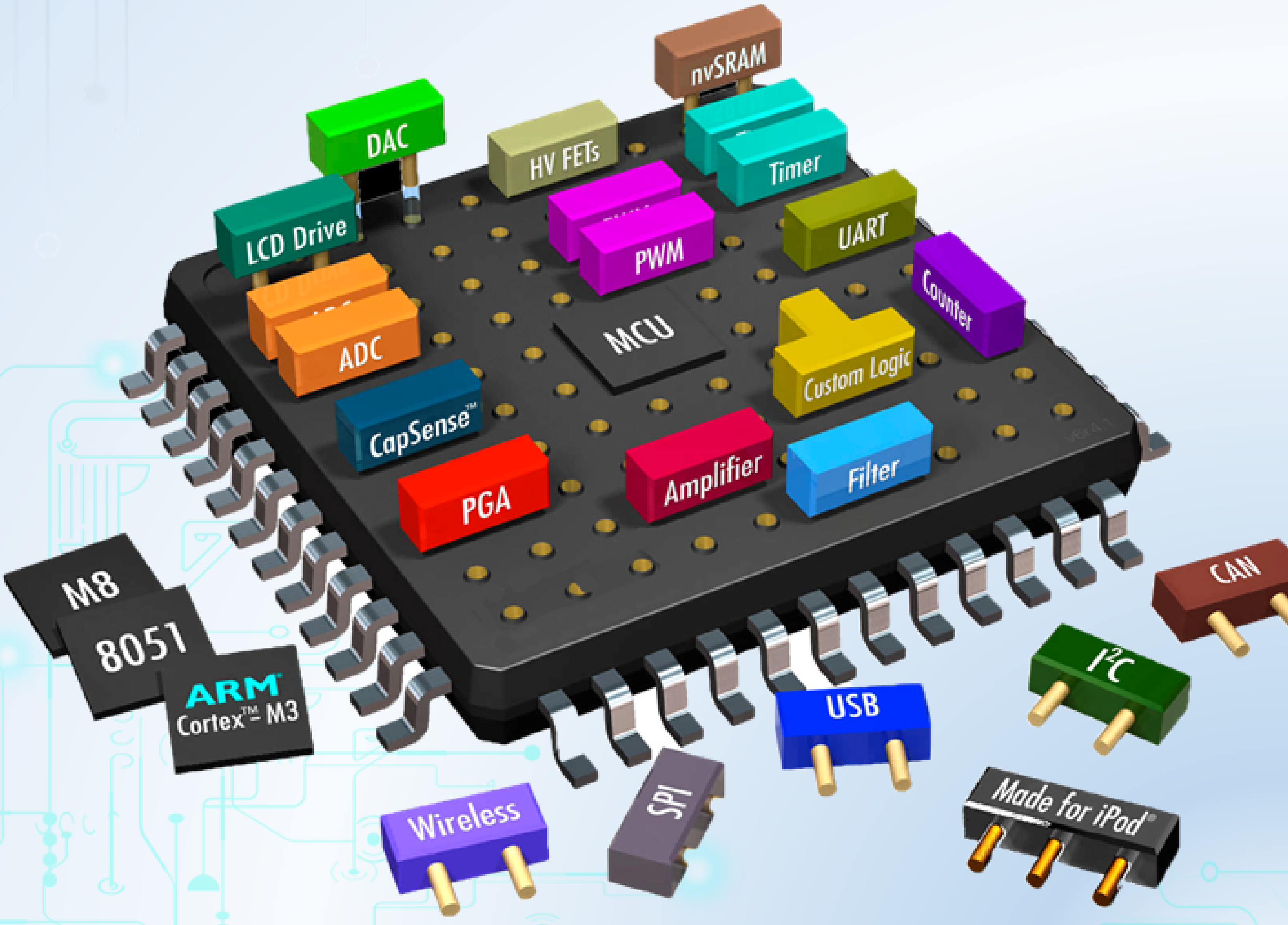
Instructor: Dr. Le Trong Nhan

Student: Nguyen Quoc Kiet - 1952802

Mục lục

CHƯƠNG 1

A Serial Communication



1 Fsm command parser

```c
#include "fsm_command_parser.h"

void command_parser_fsm()
{
  switch(status_parser)
  {
  case INIT:
    //string starts with character '!', status = WAIT_END, begins reading the command
    if(temp == '!')
    {
      status_parser = WAIT_END;
      command_index = 0;
    }
    break;

  case WAIT_END:
    // If string ends with character '#', status = INIT_STR, save the command to
    // go to analysis in uart_communication_fsm function, flag = 1.
    if(temp == '#')
    {
      status_parser = INIT;
      command[command_index] = '\0';
      command_flag = 1;
    }
    else
    {
      // If received char '!', reset command_index, reread the command
      if (temp == '!')
        command_index = 0;
      else
      {
      // Else, continue reading the command
        command[command_index++] = temp;
        if (command_index == MAX_BUFFER_SIZE)
          command_index = 0;
      }
    }
    break;

  default:
    break;
  }
}
```

2 Fsm uart communication

```

#include "uart_communication.h"
void uart_communiation_fsm(ADC_HandleTypeDef hadc1, UART_HandleTypeDef huart2)
{
  switch(status_uart)
  {
  case WAIT_RST:
    // If command has completed and command = "RST" -> status = SEND_ADC, update ADC_Value, flag = 0 and setTimer
    if (command_flag == 1)
    {
      command_flag = 0;
      if (command[0] == 'R' && command[1] == 'S' && command[2] == 'T')
      {
        // Get ADC value
        HAL_ADC_Start(&hadc1);
        ADC_value = HAL_ADC_GetValue(&hadc1);
        HAL_ADC_Stop(&hadc1);
        //HAL_UART_Transmit(&huart2, (void*)str, sprintf(str, "\r\n"), 1000);
        status_uart = SEND_ADC;
        setTimer(1, 3000);
      }
    }
    break;

  case SEND_ADC:
    HAL_GPIO_TogglePin(LED_RED_GPIO_Port, LED_RED_Pin);
    HAL_UART_Transmit(&huart2, (void*)str, sprintf(str, "!ADC=%lu#\r\n", ADC_value), 1000);
    status_uart = WAIT_OK;
    break;

  case WAIT_OK:
    // If command has completed and command = "OK" -> status = WAIT_RST and clearTimer
    if (command_flag == 1)
    {
      command_flag = 0;
      if (command[0] == 'O' && command[1] == 'K')
      {
        HAL_UART_Transmit(&huart2, (void*)str, sprintf(str, "\r\n"), 1000);
        status_uart = WAIT_RST;
        clearTimer(1);
      }
    }
    // Else, if each after 3s the system doesn't receive string "OK
```

```
   " -> status = SEND_ADC
    if(timer_flag[1] == 1)
    {
      status_uart = SEND_ADC;
      setTimer(1, 3000);
    }
    break;
  default:
    break;
  }
}
```

3 Software timer

```
#include "timer.h"

int TIME_CYCLE; // Time cycle

// Khoi tao cac mang timer_counter va timer_flag voi so luong phan tu moi mang bang NUM_OF_TIMERS
int timer_counter[NUM_OF_TIMERS] = {0};
int timer_flag[NUM_OF_TIMERS] = {0};

// setTimer (index: Timer duoc xet toi; duration: Thoi gian setTimer (don vi: ms))
void setTimer(int index, int duration)
{
  timer_counter[index] = duration / TIME_CYCLE;
  timer_flag[index] = 0;
}

// clearTimer (timer_flag[index] = 0, timer_counter[index] = 0)
void clearTimer(int index)
{
  timer_counter[index] = 0;
  timer_flag[index] = 0;
}

// timerRun
void timerRun()
{
  // Dung vong lap "for" de xet doi voi tung Timer
  // Voi moi Timer, so chu ki timer_counter se dem nguoc, neu timer_counter == 0 thi timer da dem nguoc ve 0, khi do timer_flag se bao gia tri bang 1
  for (int i = 0; i < NUM_OF_TIMERS; i++)
  {
    if (timer_counter[i] > 0){
      timer_counter[i]--;
      if (timer_counter[i] <= 0){
            timer_flag[i] = 1;
         }
      }
   }
}
```