ELEC-240 Lab4

Interfacing External Hardware to the STM32F429 Nucleo-144 Development Board

1 Introduction

This lab task covers interfacing external hardware to the GPIO pins of the STM32F429 Nucleo-144 development board. Specifically, this lab exercise focusses on driving externally connected LEDs using the GPIO pins configured as outputs.

When configured as an output the GPIO pin uses two transistors which are alternately switched to pull the output either up to VDD (3.3V) or down to GND (0V), see Figure 1. This is called Push-Pull (Figure 1a) and is the default operation mode.

Alternatively, the pin can be configured in software using the OTYPER register to operate in `Open-Drain' mode (Figure 1b). In Open-Drain mode the upper transistor is disconnected leaving only the lower (GND connecting) transistor operating.

1.1 Learning Outcomes

- By the end of this lab exercise you should be able to:

1. Demonstrate an understanding of
    a. GPIO architecture
    b. GPIO output modes and their uses
    c. Current sourcing and sinking
    d. Current limiting resistor function and how to calculate the required value
2. Produce code to configure a GPIO pin as a digital output in either mode
3. Connect an external LED to a GPIO pin in a current-source and current-sink arrangement

Task 1

From Table 2 refer to the STM32F427/429 family datasheet pin assignments section to identify GPIO pins that are not being used by other peripherals on GPIOD.

- From the Electrical characteristics section identify:

1. The maximum output current each I/O pin can sink?
2. The maximum output current each I/O pin can source?
3. The maximum total current into the VDD (3.3V power) pin?
4. The maximum total current out of the VSS (ground) pin?

Task 2

1. How much current is required to illuminate the LEDs according to their respective datasheets? (Table 2)
2. What is the maximum number of LEDs the Nucleo-144 board can drive directly from an I/O pin using this current?

Task 3

Calculate the resistor values required to provide each LED with the current and voltage specified in the datasheet when supplied from a GPIO pin and select the Nearest Preferred Value (NPV) resistor from the `E24' range available in the lab (Chart Here).

Task 4

Using the Breadboard, connect three LEDs (Red, Yellow, Green) to three GPIOD output pins configured in PUSH-PULL mode so that each LED is connected to a single port pin sourcing current.

1. Determine through experiment the largest value of resistor (or lowest current) that will still illuminate the LED? (Hint: a potentiometer is provided)
2. What is the smallest value of resistor (or highest current) that gives the brightest light without overloading the LED or the port pin?
3. Write some code to flash each LED at a different frequency.

- Make sure all code generated for the LEDs is in a file called ***LED.c*** and all functions are declared in ***LED.h*** and have "good" **comments.**

Task 5

1. Re-configure the port pins in Open-Drain mode:
   a) Do the LEDs light?
   b) If not, why? (Hint: refer to Figure 1)
2. Reconnect the LEDs so that they will work with open-drain pins (Hint: SEE HERE)
3. Re-configure the LEDs so that they're connected in series to a single port pin.

   Is it possible to illuminate all three LEDs from a single GPIO pin?
4. Is it possible to illuminate two LEDs from a single GPIO pin?
5. Find a configuration which will allow two LEDs to be lit when connected in series to single port pin.
6. What are the benefits of using open-drain pins?

Task 6

Revert the three LEDs back to the original parallel connection.

1. Connect a further three LEDs (Red, Yellow, Green) to another three GPIOD output pins
2. Write a program to control both banks of Red, Yellow, & Green LEDs to mimic traffic lights at a junction.

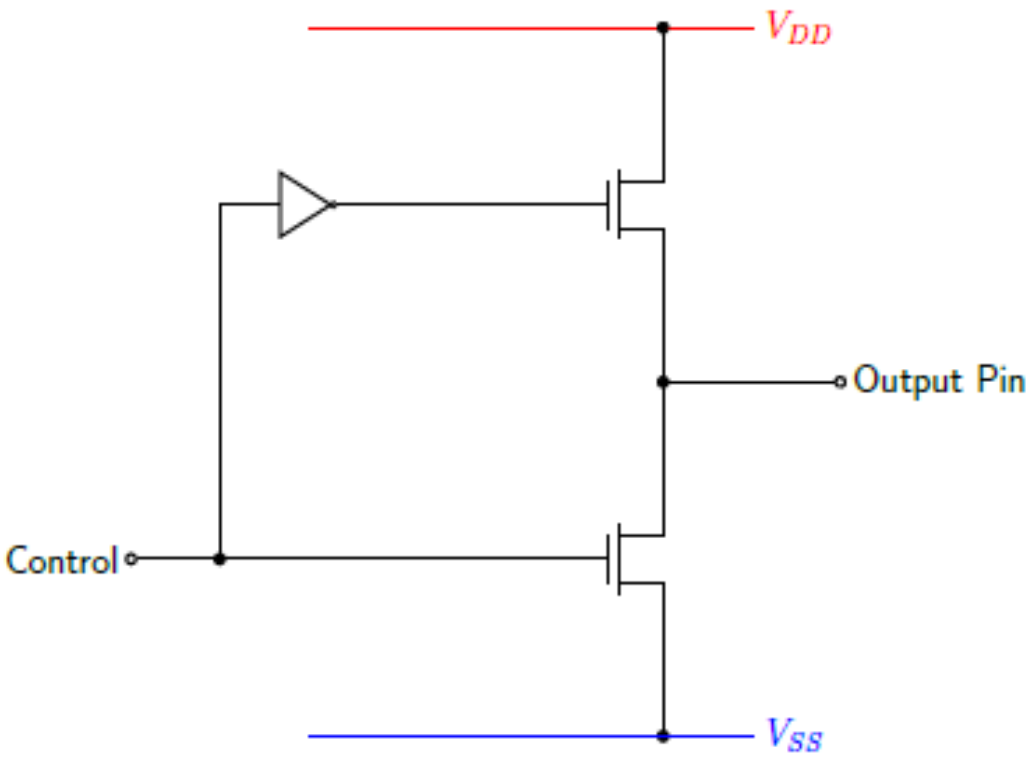
Task 7

Extend the above program to incorporate the Blue User push-button to switch between the RED and GREEN states of the traffic lights. Or to emulate a pedestrian crossing (another colour LED is provided). (You will need to refer to the UM1974 User manual to identify which pin the User Button is connected to.)

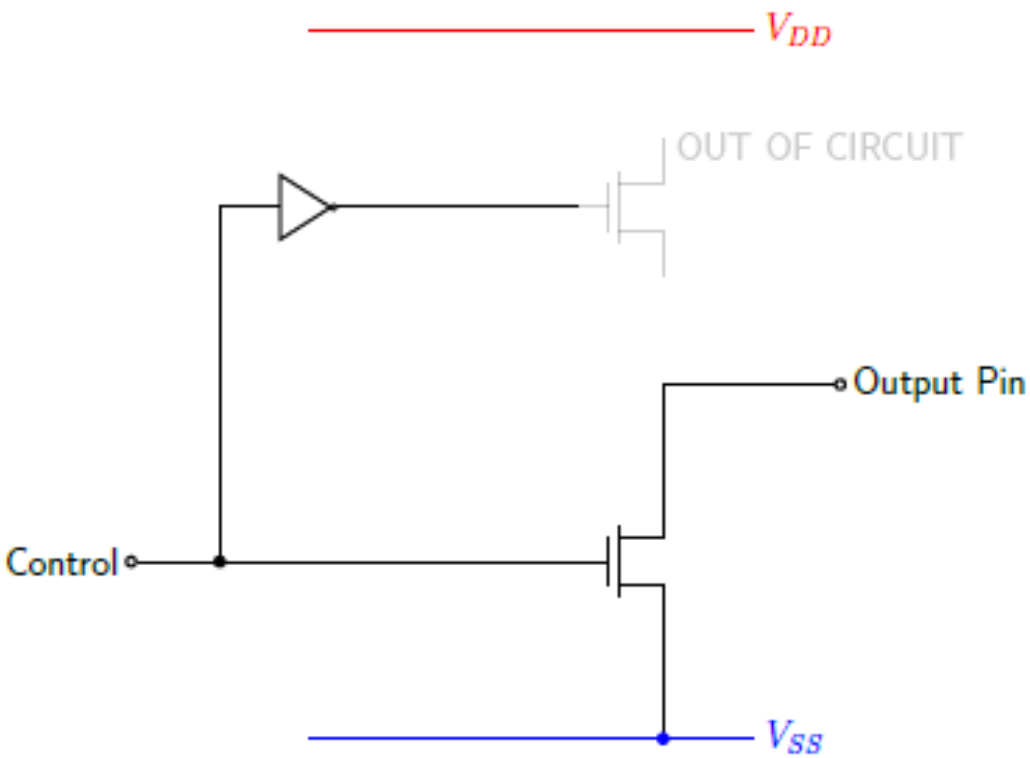
- Make sure ALL registers relating to the function of each port pin used are set in the initialisation code. **Do not assume the default condition is zero.**

Register Name	**Reference Manual Section**
Hardware Clock Enable Register (AHB1ENR)	6.3.10
GPIO port mode register (MODER)	8.4.1
GPIO port output type register (OTYPER)	8.4.2
GPIO port output speed register (OSPEEDR)	8.4.3
GPIO port pull-up/pull-down register (PUPDR)	8.4.4
GPIO port input data register (IDR)	8.4.5
GPIO port output data register (ODR)	8.4.6
GPIO port bit set/reset register (BSRR)	8.4.7

Table 1: Table of relevant control registers for configuring and controlling GPIOs



(a) Push-Pull output pin equivalent circuit



(b) Open-Drain output pin equivalent circuit

Figure 1: Output pin equivalent circuits

2.Support Documentation

Document Name	Contained Information
UM1974 User manual	• Pin identification and the supported special functions • Circuit schematics • Jumper and component identification • Header pinouts
RM0090 Reference manual	• MCU memory and peripherals architecture • Peripheral control registers, addresses and bit-fields
Green LED Datasheet Red LED Datasheet Yellow LED Datasheet White LED Datasheet	• Features • Applications • Absolute maximum rating • Electrical – Optical characteristics
E24 Resistor Chart	• Table of Standard Resistor Values
STM32F427/429 family datasheet	• Functional overview • Pinouts and pin description • Memory mapping • Electrical characteristics • Package information

Table 2: Table of relevant support documentation
(The document names are hyperlinks, please click on them to access the documents)