Embedded systems

Made by Stefan Andonov Date: 20/10/22

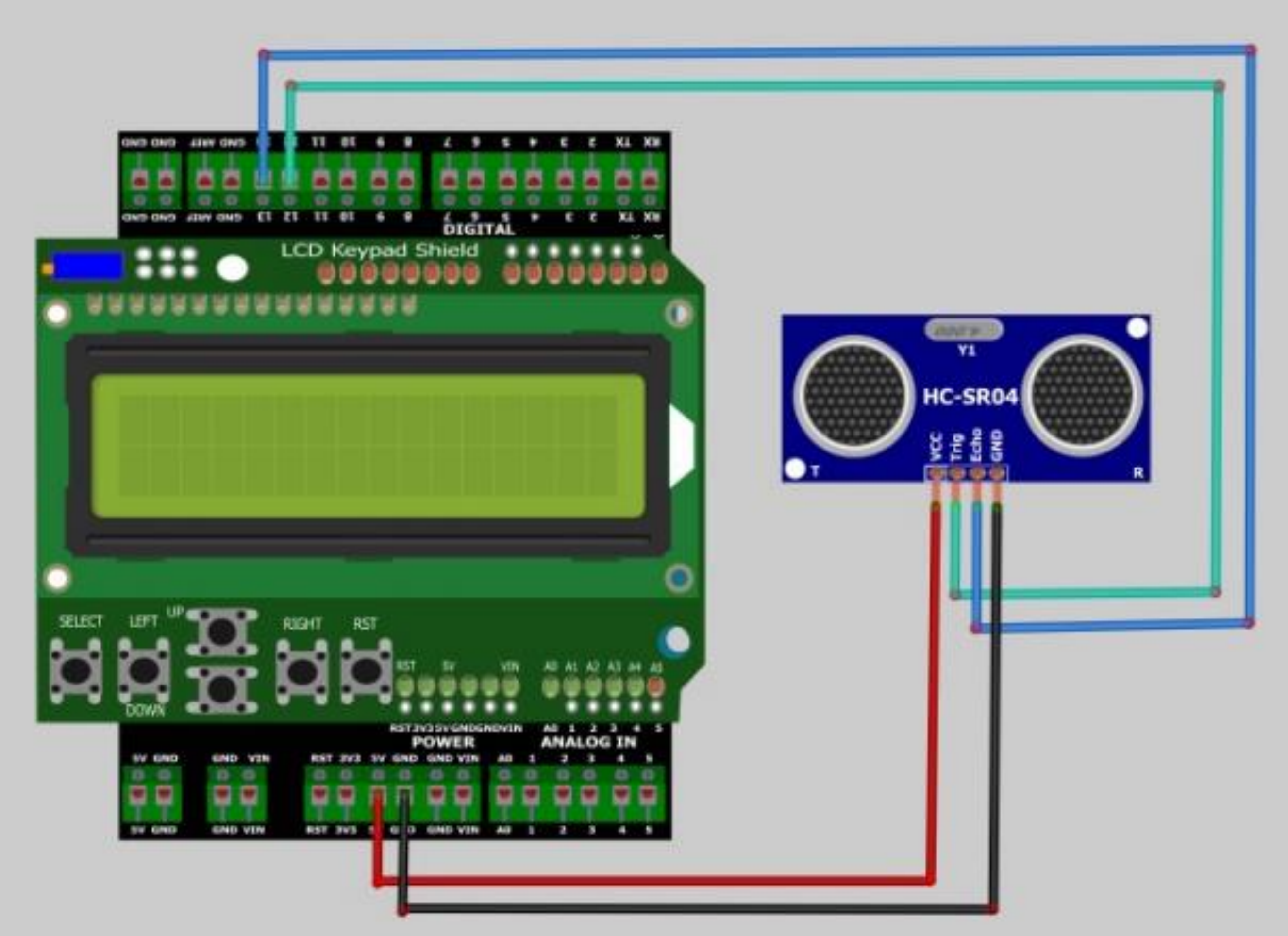
Contents

Introduction

This assignment is about utilizing the general-purpose timers of the STM32-Nucleo board in input capture mode. We are going to program an ultrasonic sensor and measure the distance to moving objects.

Procedure

The ultrasonic sensor can be used to measure distance. It has 4 pins – VCC, GND, trigger and echo. The way it works is the trigger emits a high frequency wave – 40kHz – in a burst. This wave bounces off the object in front of the sensor and the echo captures it. By measuring the time it takes for the wave to return we can deduct the distance between the sensor and the object in front.



We are going to program this by using 2 timers – one is going to be setup as output – for the trigger, and the other will be in input capture mode – the trigger. We are using TIM3 for output and TIM4 for input capture, therefore we are going to connect the trigger to pin D4 and echo to pin D10 on the STM32-F303RE board.

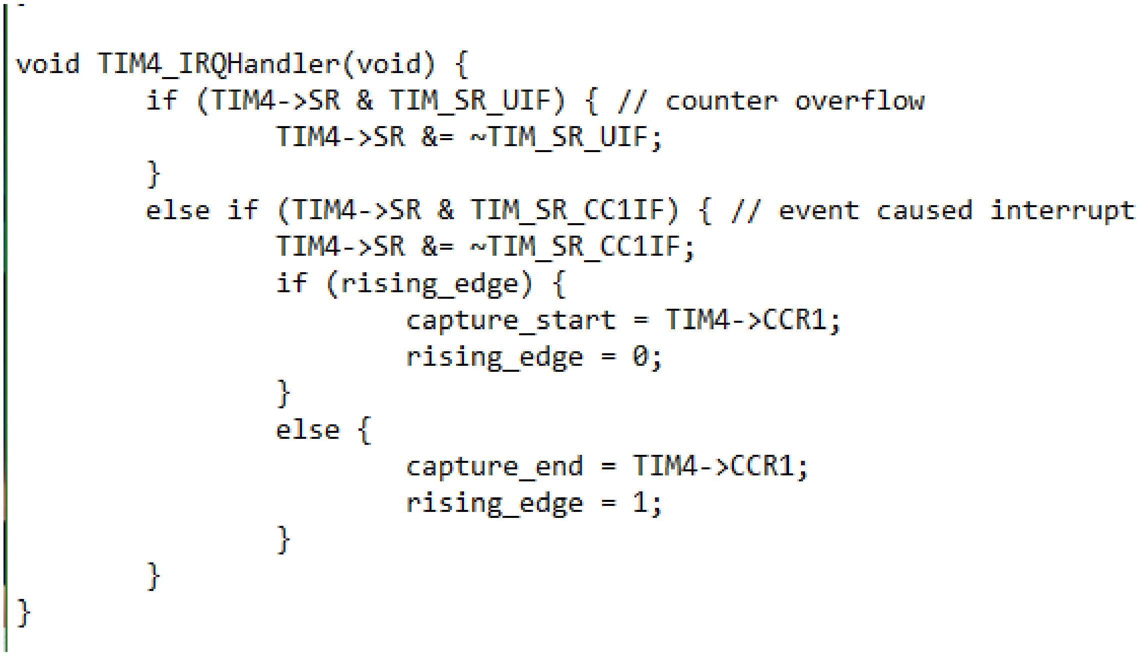
Sensor	Pin	Alternate Function
Trigger	PB 5	TIM3_CH2
Echo	PB 6	TIM4_CH1

We are going to setup TIM3 in PWM mode 1 and set the trigger to a high state for 10uS every 65ms. We achieve this by using the highest possible auto reload value (0xFFFF) and slowing down the timer to 1mHz by setting the pre-scaler to 15. Our internal clock is set to 16mHz in this example.

TIM4->ARR	0xFFFF
TIM4->PSC	15
Clock speed	16mHz

We are going to setup TIM4 in input capture mode and we are going to select 0 filtration and pre-scaling. This means that only 1 sample of HIGH->LOW or LOW->HIGH will mean we have a valid event. We are also using no pre-scaling because we want to generate an interrupt on each falling and rising edge event.

In the input timer interrupt handler we are going to use a global variable “rising_edge” to determine which edge caused the interrupt to be generated.



```
void TIM4_IRQHandler(void) {
        if (TIM4->SR & TIM_SR_UIF) { // counter overflow
                TIM4->SR &= ~TIM_SR_UIF;
        }
        else if (TIM4->SR & TIM_SR_CC1IF) { // event caused interrupt
                TIM4->SR &= ~TIM_SR_CC1IF;
                if (rising_edge) {
                        capture_start = TIM4->CCR1;
                        rising_edge = 0;
                }
                else {
                        capture_end = TIM4->CCR1;
                        rising_edge = 1;
                }
        }
}
```

We are going to then measure the distance with the following calculation:

distance_cm = (capture_end – captue_start) / 58;

Now that we have the distance, we are going to use it as output on a PWM pin, to change the brightness of an LED. The closer you are to the sensor, the more the LED brights up. This is achieved with the PWM mode 2 of the LED. You can find a demonstration of the entire project on the following link:

Timers input capture Ultrasonic sensor

References

Technology / t-sem3-cb · GitLab (fhict.nl)