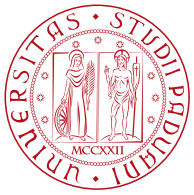

Università
degli Studi
di Padova

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Distance Sensor

ICT for Industrial Applications 2022/23 Final Project

Authors: Amerigo Aloisi, Paolo Bettin, Luca Bonaventura, Jaswant Singh Bogan
Supervisor: Andrea Zanella

19/06/2023

Contents

1 Introduction

The goal of this project is to implement a distance measurement system using a ultrasonic sensor, allowing it to be used in applications that require proximity sensing, object detection, or obstacle avoidance. By accurately measuring distances, the system provides data that can be used for making informed decisions or triggering appropriate actions.

2 Setup description

All the code as been developed and tested on Infineon's **Aurix Kit TC397 TFT**. The following on-board modules have been used.

2.1 GTM Module

A powerful general purpose timer, used for measurement of the elapsed time and generation of the PWM signal for the buzzer. The choice of using it allows to save CPU cycles, since the CPU has to only read the ticks, and not constantly count them. This is an important aspect that must be considered when working with embedded platforms. Timer resolution up to 24 bits with up to $10ns$ time granularity. The module works at 100 MHz but his frequency can be changed setting the value of GTMDIV [1]. This could be useful if the 24 bit counting register [2] is saturated. The largest measurable time equals:

$$T_{max} = \frac{2^{24} - 1}{100MHz} = 167.72ms$$

which is enough for the purpose of this project.

2.2 Others

- Buzzer: simple audio buzzer.

In addiction, two devices have been connected to the board:

- HCS04 Ultrasonic sensor: this sensor bursts an ultrasound signal and capture the reflected signal back.
- LED Matrix: the driver uses SPI to perform synchronous communication.

3 Implementation

The measurement process is done inside the `DistanceSensor.c` file. First the modules and devices are initialized with the `initPins()` function. The ultrasound sensor has two I/O PINS: Trigger and Echo. Trigger is connected to a GPIO port set to output mode, the PIN needs to be set high for 10 μs so the sensor will burst a signal. After the burst the Echo pin of the sensor is automatically set high and it will stay in that state until the reflected signal is captured back by the sensor or until the timeout expires ($38000\mu s$). The Echo pin is connected to the input pin of the GTM module. In this way, the timer will count for all the time elapsed until the reflected signal is detected and after that the pin is automatically set low again. From the number of timer ticks the distance of the object hit by the signal can be found with the following operation:

$$d = \frac{v_{sound}}{2} \cdot T_{Tick} \cdot N_{Ticks} \cdot 3$$

The multiplication by 3 is caused by the fact that the timer operates at 100Mhz while the CPU works at 300Mhz. This operations are done inside `triggerSensor()` function.

[1] `https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part1-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b701717d3605221d96`

[2] `https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part2-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b701717d35f8541d94`

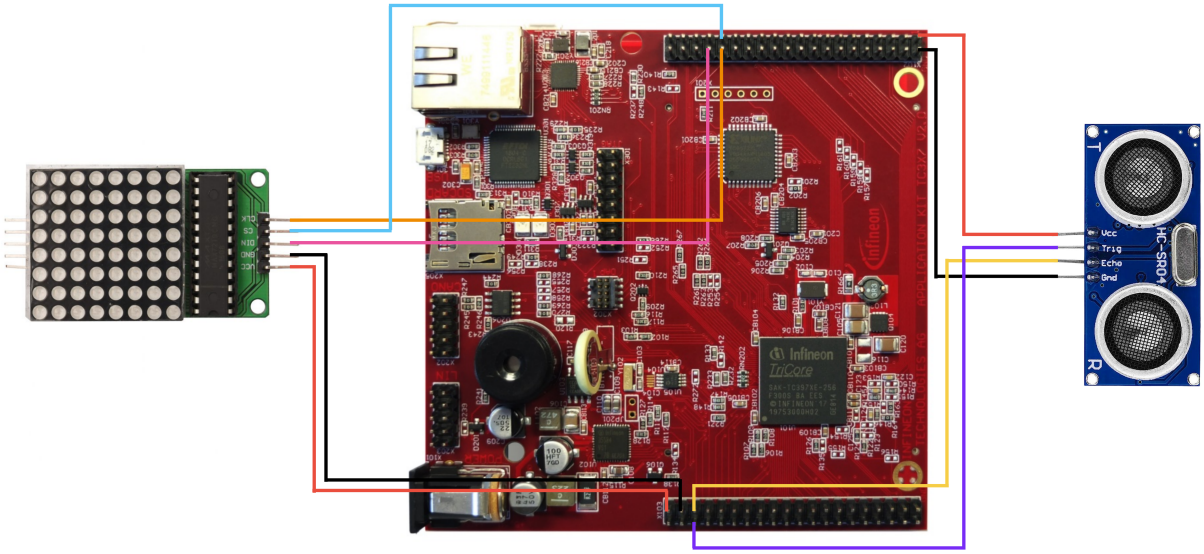

Figure 1: Wiring diagram

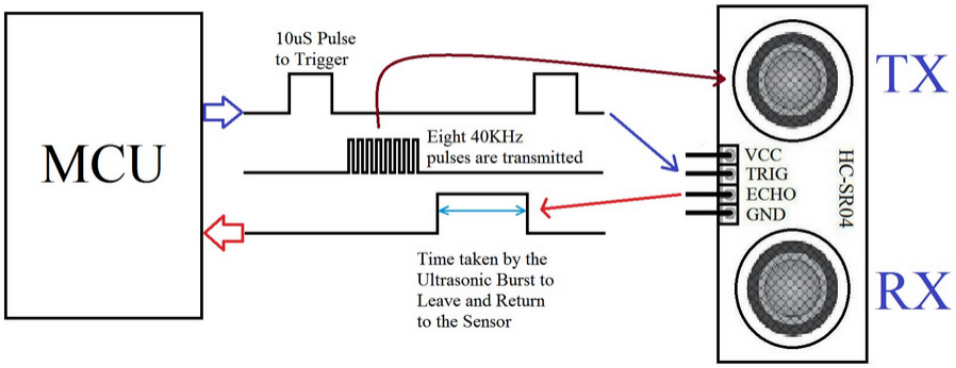

Figure 2: HCS04 working system

The `PWM_Buzzer.c` controls a audio buzzer using Pulse Width Modulation (PWM) signals generated by the GTM. It includes functions for initializing the PWM configuration, setting the volume of the buzzer, and playing tones. The code initializes the GTM module, configures the TOM Timer for PWM generation, and provides functions to control the volume and play tones on the buzzer.

The `SPI_CPU.c` file contains function to setup and manage the SPI connection between the board and the LED matrix. In particular, the `printLedMatrix(int range)`'fuction is responsible for displaying the measured distance on the LED matrix. The implementation of this function include the following steps:

- Mapping Distance to LED States: The measured distance value s is mapped to the LED states in order to determine which LEDs should be turned on or off.
- LED Matrix Update: The function updates the LED matrix display based on the mapped LED states, illuminating the appropriate LEDs to represent the distance visually.
- Visual Representation: depending on the specific application requirements, the function may employ various patterns, symbols, or animations to convey the distance information effectively. This could include displaying bars, digits, or graphical icons to indicate the distance level. Bars in a "WiFi like" style have been chosen for this implementation. By integrating the LED matrix display, the system provides a visual indication of the measured distance, enhancing the user experience and allowing for intuitive interpretation of the distance values.

Finally, the `Cpu0_main.c` file contains the initialization and main loop for CPU core 0 in the system. It enables interrupts, disables watchdogs, and initializes various peripherals and modules. The code initializes the pins, the TIM module for capturing PWM signals, and the SPI communication for LED matrix. Inside the loop it performs the distance measurement, displays the distance on the LED matrix and plays tones on the PWM buzzer if the object is close. The code provides an integrated solution for distance measurement, LED matrix display, and PWM buzzer functionality.

4 Issues

The project was initially meant to be developed on **AURIX Kit TC375 Lite**. This first board was not able to output enough power for both the sensor and the LED Matrix, so it had to be changed with the more powerful AURIX TC397 TFT. Moreover, again for the power requirements, the Aurix board had to be powered with an external power supply set to 7.5V rather than powering it through the 3.3V micro-USB port connected to the computer.