1. Project title: Speed meter

Ungureanu Sebastian

Grupa 1B

tepan9415@gmail.com

Grigoriciuc Bogdan

Grupa 1B

grigoriciucbogdan@gmail.com

2. Summary

In this project, we implemented a method to measure the speed of an object (a ball). For this purpose, we used 2 motion sensors situated at a distance of 2 meters one from the other. We need to know in what time the ball covers the distance in order to calculate the speed of it. The speed is calculated with the following formula: v=d/t (m/s).

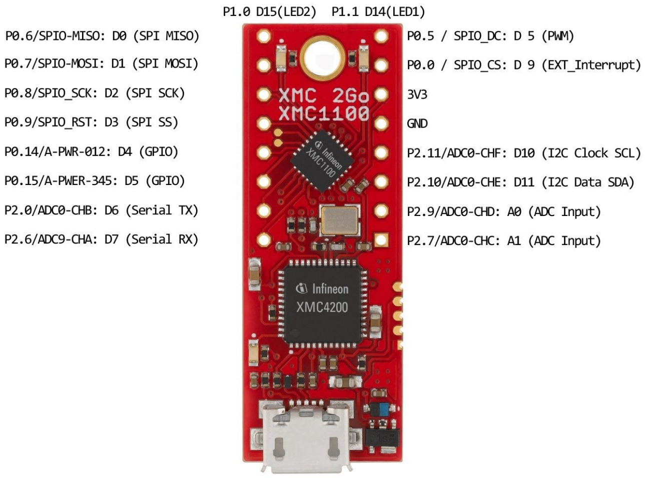
3. Introduction, project goals

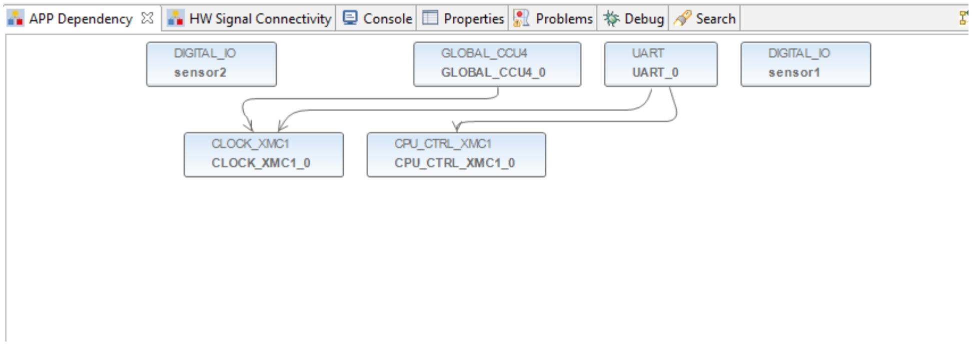
By working on this project we wanted to find a simpler method of calculating the speed of a moving object. For this, we used the xmc 1100 2go microcontroller connected to a mobile phone through a bluetooth module. The calculated speed is, then, submitted to the terminal on the mobile phone.

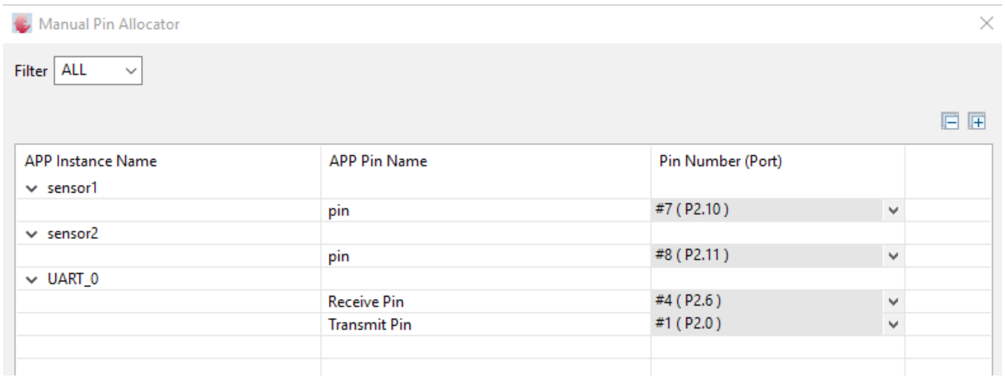
4. Components

For the project we used the following components:

- Xmc 1100 2go


P1.0 D15(LED2)
P1.1 D14(LED1)
P0.6/SPIO-MISO: D0 (SPI MISO)
P0.7/SPIO-MOSI: D1 (SPI MOSI)
P0.8/SPIO_SCK: D2 (SPI SCK)
P0.9/SPIO_RST: D3 (SPI SS)
P0.14/A-PWR-012: D4 (GPIO)
P0.15/A-PWER-345: D5 (GPIO)
P2.0/ADC0-CHB: D6 (Serial TX)
P2.6/ADC9-CHA: D7 (Serial RX)
P0.5 / SPIO_DC: D 5 (PWM)
P0.0 / SPIO_CS: D 9 (EXT_Interrupt)
3V3
GND
P2.11/ADC0-CHF: D10 (I2C Clock SCL)
P2.10/ADC0-CHE: D11 (I2C Data SDA)
P2.9/ADC0-CHD: A0 (ADC Input)
P2.7/ADC0-CHC: A1 (ADC Input)
XMC 2Go
XMC1100
Infineon
XMC4200




Manual Pin Allocator

Filter ALL

APP Instance Name	APP Pin Name	Pin Number (Port)
sensor1		
	pin	#7 (P2.10)
sensor2		
	pin	#8 (P2.11)
UART_0		
	Receive Pin	#4 (P2.6)
	Transmit Pin	#1 (P2.0)

- 2 motion sensors
- wires

- usb cable
- bluetooth module
- Dave as IDE
- Bluetooth terminal

DAVE (Infineon) Digital Application Virtual Engineer (DAVE™), a C/C++-language software development and code generation tool for microcontroller applications. DAvE is a standalone system with automatic code generation modules and is suited to develop software drivers for Infineon microcontrollers and aids the developer with automatically created C-level templates and user desired functionalities.

Latest releases of DAVE include all required parts to develop code, compile and debug on the target for free (based on the ARM gcc toolsuite). Together with several low-cost development boards one can get involved in microcontroller design very easy. This makes Infineon micocontroller products also more usable to small companies and to home-use / DIY projects - similar to established products of Atmel (AVR, SAM) and Microchip (PIC, PIC32) to name a few.

DAVE was developed by Infineon Technologies. Therefore, the automatic code generator supports only Infineon microcontrollers. The user also has to get used to the concept of the Eclipse IDE. The generated code can be also used on other (often non-free) development environments from Keil, Tasking and so on.

Active Infrared Beam Detectors (IR Beams) adopt the invisible infrared detection technology. It consists of a receiver and a transmitter, the transmitter emits 2-4 infrared beam, and the receiver will receive the infrared beam.

Infrared Fence Beams sensors are effective and unobtrusive methods for greatly increasing the level of protection at your home or business. Based on infrared technology, they can be configured to monitor the perimeter, driveways, swimming pools and unauthorized areas. These systems detect both people and vehicles and as such, are extremely versatile.

Other projects using motion sensors:

- Motion alarm
- Automatic door
- Camera trigger.

5. Source code

```
#include <DAVE.h>            //Declarations from DAVE Code Generation (includes SFR declaration)
#define oSec 0x17222f                        // numarul de incrementari pentru a estima o secunda
int main(void)
{
        DAVE_STATUS_t status;
        uint32_t i,cnt=0;
        status = DAVE_Init();       /* Initialization of DAVE APPs  */
        float t,v;
        uint32_t IR1, IR2;
```

```
    uint8_t in,zec,flag=0,d=2;
    if(status != DAVE_STATUS_SUCCESS)
    {
        while(1){}
    }
    while(1)
    {
        IR1 = DIGITAL_IO_GetInput(&sensor1);
        IR2 = DIGITAL_IO_GetInput(&sensor2);
        if(IR1==0 && flag == 0)
        {
            flag=1;
        }
        if(flag==1)
        {
            cnt=cnt+1;
        }
        if(IR2==0 && flag == 1)
        {
            t=(float)cnt/oSec;
            v=d/t; // calcul viteza
            v=(int)(v*100);
            zec=(int)(v)%100;
            v=v/100;
            in=(int)(v)%255;
            UART_Transmit(&UART_0, &in,1); //trimit partea intreaga
            UART_Transmit(&UART_0, &zec,1);//partea zecimala (2 zecimale in acest caz)
            UART_Transmit(&UART_0, "\n",1);
            cnt=0;
            flag=0;
        }
    }
}
```