02_Ultrasonic Obstacle Detection

Introduction:

In this project there is a need to detect the nearest obstacle in front of the robot. In order to realize this functionality we decided to use an ultrasonic sensor. The model used here is the HC-SR04. This sensor has four pins: a Vcc pin, a Gnd pin, a Trig pin and an Echo pin.

Sub-tasks:

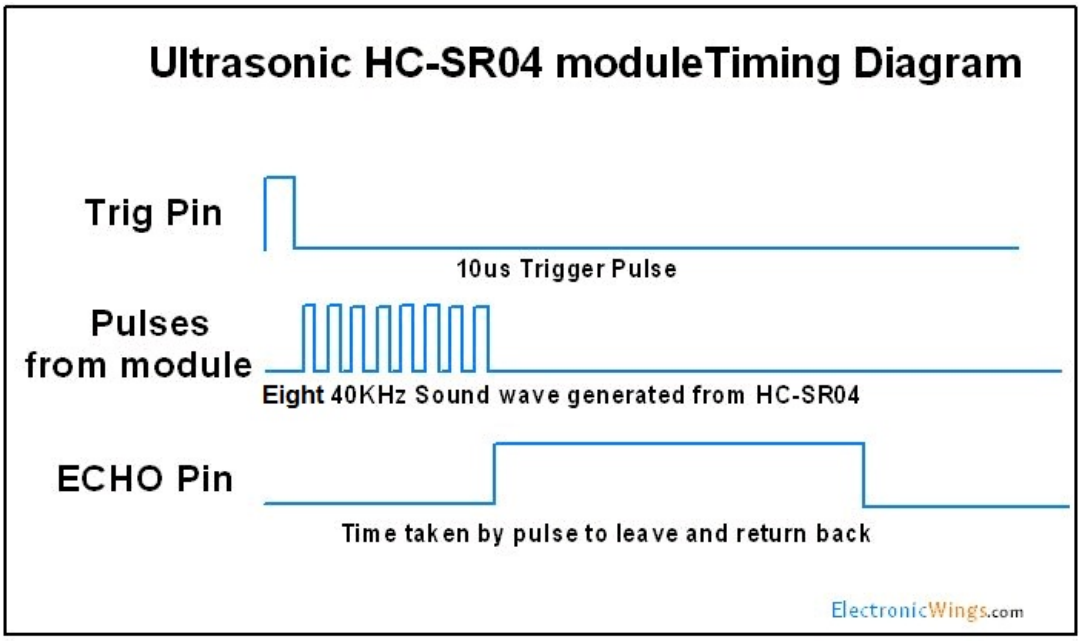
For this particular kind of sensor to be configured and working properly, three main tasks need to be done:

1. Configure an external interrupt using the Echo pin as an input to receive the signal equivalent to the distance of the obstacle

2. Send a 5V signal that lasts at least 10 µs on the Trig pin to trigger the measurement

3. Process the information provided by the Echo pin to compute the distance to the obstacle

Here is the timing diagram for the HC SR04:



Sub-tasks implementation:

We will now define how are realized each sub-tasks.

1. To configure the external interruption what is needed is the External Request Unit(ERU) module of the AURIX™. This module allows us to use some dedicated pins of the AURIX™ as input for external signal that will be the cause of the interrupt.

In our case the pin used will be the P20.9. The list of possible input for the AURIX™ used here (TC29xB) can be found in the user manual under ERU Input Pin Connections. We can see that this pin is linked to the ERS7 and corresponds to the input In70. This also means that the ERU used here will be ERU3. The reason why that is so is because there are 4 ERU in this AURIX™: ERU0 to ERU3. Each of those are responsible for 2 ERS, for example ERU0 taking care of ERS0 and ERS1, and ERU3 taking care of ERS6 ans ERS7. Those ERU are not responsible only for those ERS but that is not what matters for us as of now.

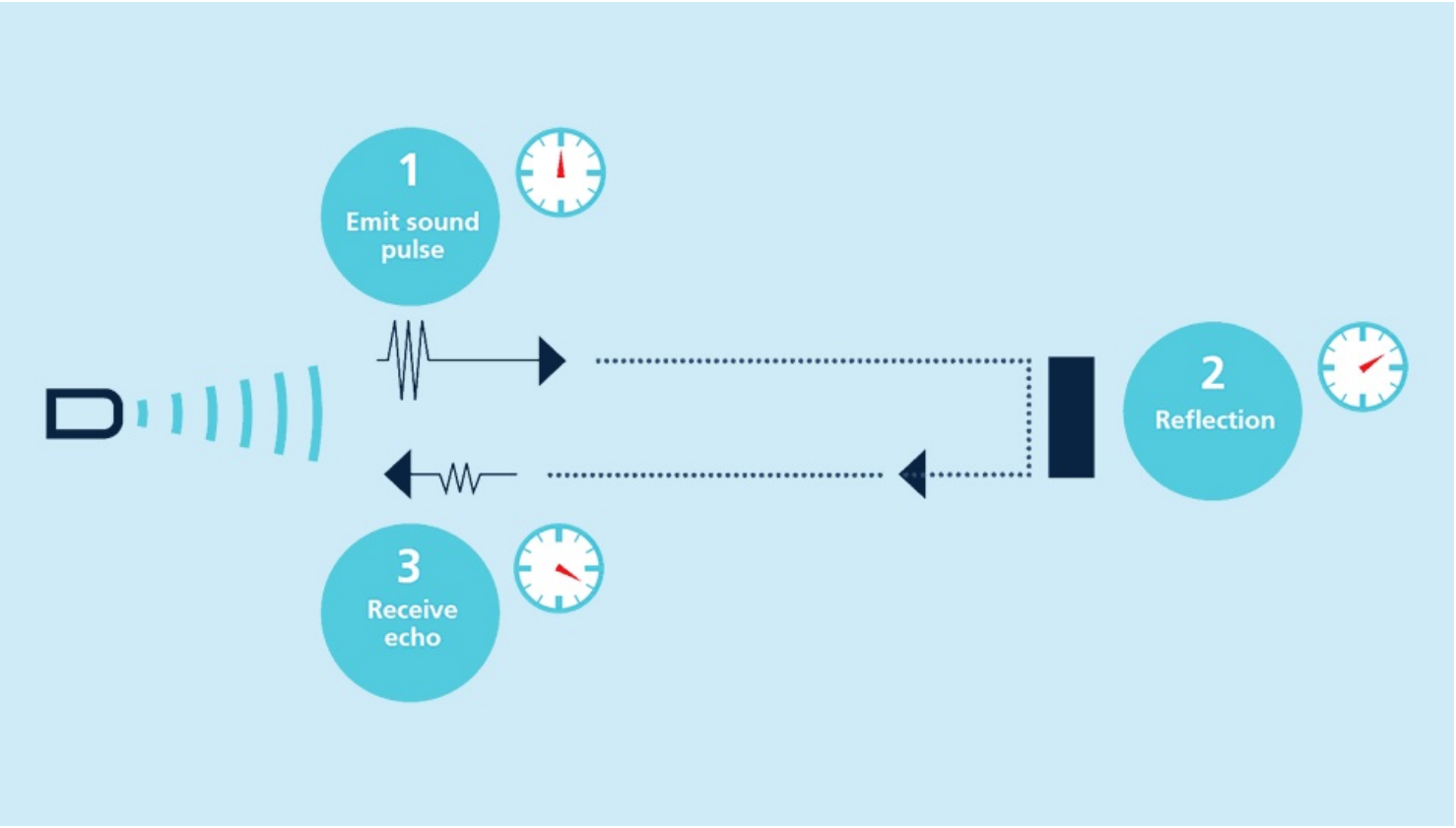
Now that we know which ERU and which ERS input to consider we can take a look at the External Input Channel Registers EICRi (i=0 to 3). These registers are the ones needed to configure the pattern which to look for in the signal given as input to the ERU. In these register there are two big parts, the parts to configure input for which the first number corresponds to (2i) or (2i+1), i being the index of the EICRi. For example input In70 corresponds to EICR3 as 23+1 = 7. As such according to the user manual the bitfield to be configured are SCU_EICR3.B.BITFIELD0 if first number of input is (2i) or SCU_EICR3.B.BITFIELD1 if (2*i+1) where BITFIELD is the name of the bitfield. There is a need to know where will the interrupt output go through. To see which Output Gating Unit OGU to configure one need to look through ERU Output Pin Connections and find the OGU which corresponds to the ERU used, here there are two possibilities either OGU3 or OGU7.

What needs to be configured then is the Interrupt Gating Register IGCRi (i=0 to 3), regarding the behavior of the interrupt output.

2. To start the detection, a signal of at least 10 µs need to be generated. To do so, a function using System Timers STM is useful. This function is the IfxStm_waitTicks, that provided the number of ticks and the STM used can generated a controlled pause. Thus setting the pin to a high state at the beginning then after the pause setting it to low.

3. When a signal arrives on the Echo pin it triggers an interrupt. This interrupt is set so that at first it triggers on the rising edge and store the time of the rising edge, but then changes to detect falling edge and store the time it happened. This is done by the XOR on these bitfields in the isrGetDistance() function: `SCU_EICR3.B.REN1 ^= 1; SCU_EICR3.B.FEN1 ^= 1;`

At last there is a function that returns the distance by returning (tf - t0)*17/100. Knowing that the speed of sound is 340 m/s and that the sound wave is reflected back as shown in the image below, we have distance = 34 000 cm/1 000 000 µs * time_trig µs * time_delta / 2. As such when simplified we have d = time_delta * 17/100 = (tf- t0) * 17/100



Remark: As there are some times absurd results returned by this function, a test in the main before using the value is recommended.

IMPORTANT NOTE The Ultrasonic sensor is using an external interrupt going in Pin 20.9. After developing the software and integrating the project, we realized that in the application kit, P20.9 is also mapped to the Ethernet module, with a pull down resistor in between. This means that by default, the AURIX™ does not receive any information from the ultrasonic sensor because the pin is pulled down. Unfortunately the number of pins we can use for external interrupts that are accessible in the application kit is very limited. We were not able to use another pin.

There are 2 fixes to this problem, one permanent and one temporary. - The temporary fix is to do a warm Power On Reset. Power supply the robot, before enabling mode 2, click on the Power On Reset user button on the back of the application kit. What will happen is that the reset instruction will arrive to the Ethernet module after it was received by the Cpu. This will allow the AURIX™ to drive this pin before the Ethernet module is able of doing so.

- The permanent fix is to remove the pull down resistor for the PCB. The resistor is labeled as R238. It is located on the back of the application Kit (the face with the screen). You will find it on the top left of the screen. If it is not removed yet, the green part of the plastic screen protector is right in front of it. One other marker to find this resistor is the Infineon logo drawn on the PCB, R238 is located on its right, very close to it. Removing R238 does not have any impact on the usability of the application kit for the robot project. It might on the other cause a problem for using the Ethernet module.

written by Marlon Bourgogne 14/09/18

modified by Marlon Bourgogne 04/02/19

important note added by Amine Gaizi 17/06/19