

---

# Proximity alert system

Lorenzo Chelini matricola s????? - 13 dicembre 2015

---

## Introduction

The aim of this application, based on Micrium Os 3, is to set up a real time-algorithm that measures the distance of an object, between 2 to 400 cm using HC-SR04 as distance sensor and the Freescale FRDM-K64F as board.

## Code structure

The entry point is the main, it calls same function provided by the Micrium operating system in order to initialize the hardware, the GPIO pins and the operating system itself. The main function also install an handler for the PTB09 pin and create an switching semaphore with an initial value of 0.

Moreover it creates three different tasks, AppTaskStart, AppTaskLed and AppTaskBlink respectively, that will be explained in the next section.

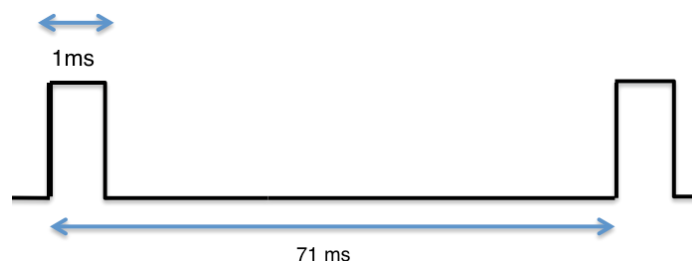
## Task

The AppTaskStart is responsible for :

1) initializing the low power timer module, calling the function `timer_ini()`.

Then inside an infinite loop these action are performed:

2) Trigger the distance sensor according to the following time diagram :



Note that between two consecutive pulse, a laps of time greater or equal to 60 ms should elapse and at least each pulse should stay high for 10 us.

- 3) Compute the new distance according to the formula of Weighted Moving Average, and update the global variable *distance*.
- 4) post on the switching semaphore to wake up the appTaskLed.

### AppTaskLed :

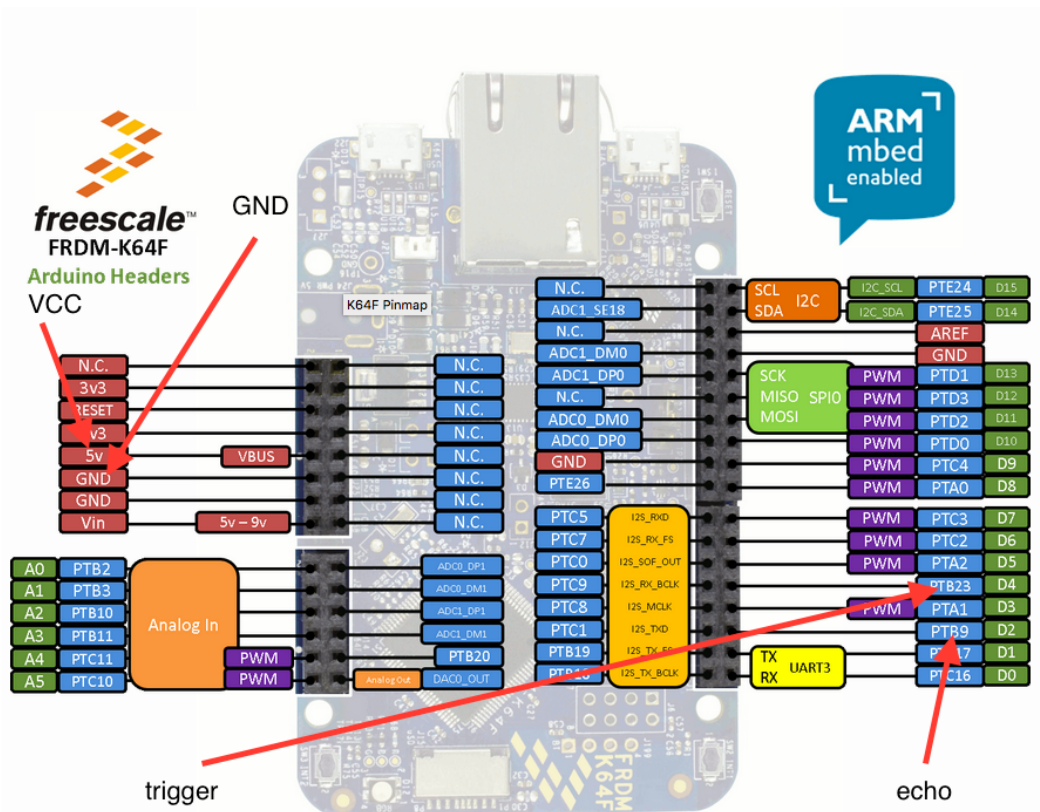
The task inside an infinite loop makes the action below :

- 1) Makes a first pend on the switching semaphore, waiting for the post computes by the AppTask.
- 2) Compute a search with complexity in the worst case  $O(n)$  in the *lower\_bound\_array*, that contains for each position, a values in centimeters that corresponds to the right bound for each range of distances. This search is done in order to verify if the new distance belongs to a new range and if so the AppTaskBlink, the task that switch on and off the led, should be resumed; otherwise nothing is required. Note that if the distance falls in a brand new range the global variables *period* and *color* are update, in order to do so, an access to the colors and periods array should be done, with the index previously computed throughout the search.

### AppTaskBlink

The task switch on the proper led, with the proper frequency, using the function provided by the operating system.

## Pin Assignment



---

outPTB23 used by the AppTaskStart to trigger the distance sensor.

inPTB9 used to receive the echo from the sensor.

Note that every time an event occur on the pin inPTB9 (both rising edge and falling edge ) an handler is lunch, in particular we are referring to handler\_PT9.

This handler is responsible for understanding if the event is a rising edge or a falling one and accordingly to this execute the proper action, in particular if we are dealing with a rising edge the handler set the TEN (timer enable) bit in the control register CSR allowing the low power timer to start counting.

On the other hand if a falling edge has been detected the value proportional to the duration of the echo is copy from the CNR register to the global variable CNR\_value and the time is reset clearing the TEN bit in the CSR register.

## Low power timer settings

In this section the low power time module configurations will be explained.

These steps are required in order to use the timer:

- 1) Set the clock for the LPTMR, in this case the fast internal reference clock (4Mhz) has been selected. (SIM\_SCGC5)
- 2) Reset the module to clear any flag/values (LPTMR0\_CSR)
- 3) Set the compare value (LPTMR0\_CMR), in these case we set the register to 0 because the timer is used in the free-running mode and not as a comparator.
- 4) Set the prescaler, in this case we by-pass it using the instruction LPTMR->PSR |= 1<<2
- 5) Enable the LPTMR
- 6) Read the content of CNR and reset the module.

## User Guide

In order to use the application the user should follow, carefully, these steps:

- 1) Open IAR Embedded Workbench IDE
- 2) Select the proper workspace in File -> Open Workspace
- 3) Connect the distance sensor:
  - 1) The echo should be connected to outPTB23
  - 2) The Trigger should be connected to inPTB9
  - 3) The VCC should be connected to +5V board pin
  - 4) The GND should be connected to GND board pin