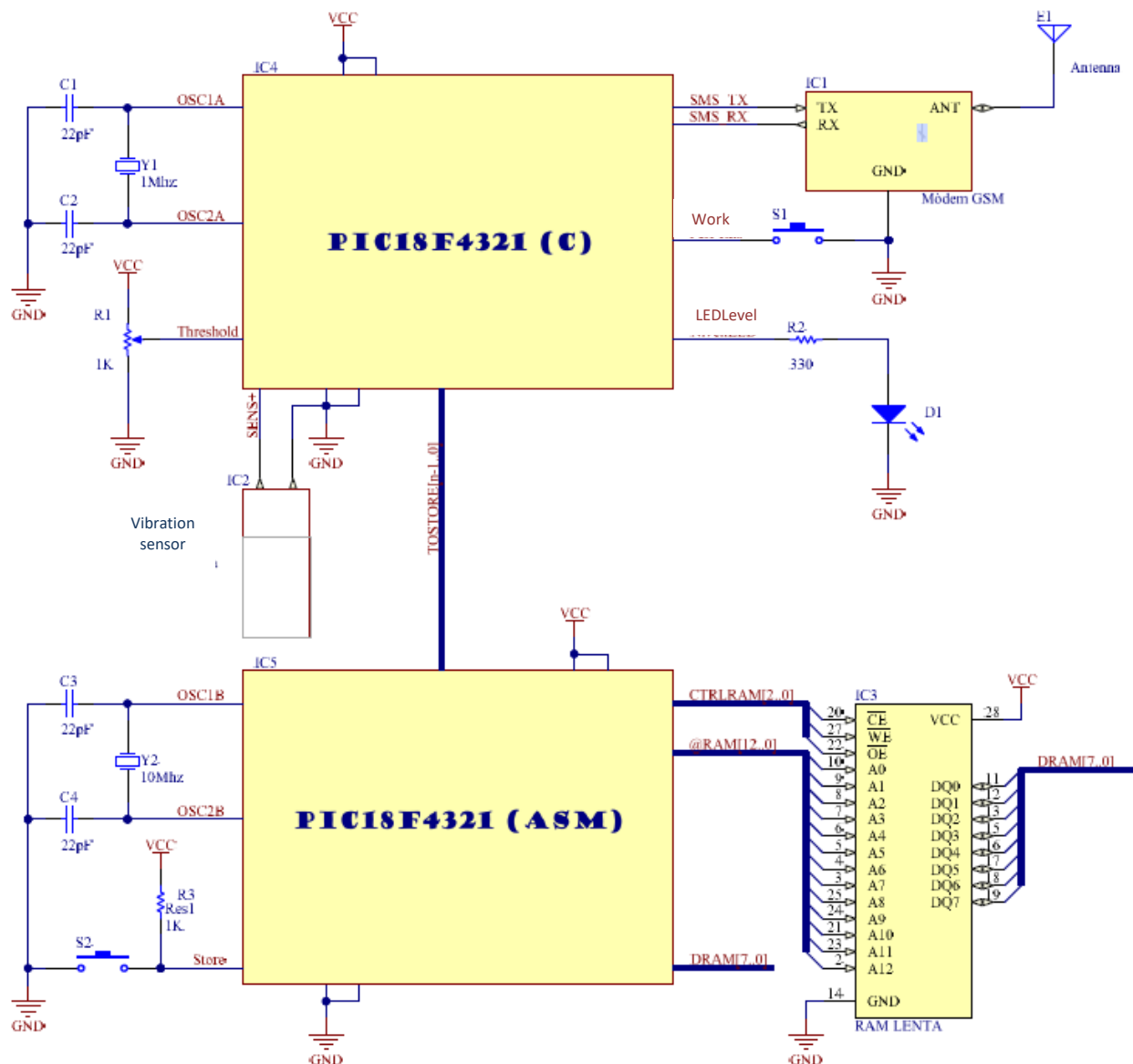


The earthquake digital detector

We want to build a system that measures and registers the subsoil vibrations to detect earthquakes. Concretely, we want this device to (1) send an SMS (Short Message Service) message to a fixed phone number when the subsoil vibrations surpass a certain limit (in other words, when an earthquake is detected), (2) control at real time the intensity of an LED diode that blinks depending on the vibration level captured, and (3) store the vibration level into an external memory. To do that, the following schematic is proposed:



Specifications:

- **Subsoil vibrations control.** To read the subsoil vibrations, we will use a piezoelectric vibration sensor. It will send to the input SENS+ a signal ranging from 0 to 5V that must be sampled at a 2KHz rate (a sample each 0.5ms.). This signal SENS+ must be compared to the analog value given by the potentiometer R1 connected at the Threshold input. If the SENS+ tension is equal or greater than the value read from the Threshold, it means that an earthquake has been detected.
- **The GSM modem.** The GSM modem (IC1 at the schematic) will be used to send an SMS every time an earthquake is detected. When an earthquake is occurring, a new SMS message must be send every 30 seconds. The system also needs to send the SMS when the pushbutton `Work` (with bounces) is pressed. The microcontroller will communicate with the modem using the RS232 protocol at 1800 bauds (signals `SMS_TX` & `SMS_RX` at the schematic). To send an SMS trough the modem, the microcontroller must generate the following AT command sequence:
 1. `"AT\r"`
 - This command is used to make sure the modem is connected. The modem must answer with a `"\r\nOK\r\n"` before 300ms.
 2. `"AT+CMGS=600696969\r"`
 - This command is used to indicate the modem we want to send a text message to the number 600696969. The modem must answer with a `"\r\nOK\r\n"` before 300ms. If it does not do it, all the command sequence must be restarted.
 3. `"A earthquake is occurring at XX:YY with value ZZZZ\r"`
 - This is the SMS text to send: `XX:YY` are the hours and minutes that have passed since the program started running, `ZZZZ` is the digital value that the vibration sensor is reading at this moment. The modem must answer with an `"\r\nOK\r\n"` before 3 seconds. If it does not do it, all the command sequence must be restarted.
- **The LED.** At all times, the LED intensity must be regulated (the signal `LEDLevel` at the schematic) proportionally to the vibration level read by SENS+. To do it, we will use the classic PWM signal at 50Hz (our generosity knows no limits).
- **Vibration level register.** Every time an earthquake is detected or when the `Store` pushbutton (without bounces) is pressed, the vibration levels 200ms before an earthquake happens (400 samples) must be stored to an external RAM (IC3 at the schematic). The data must be stable during 40 μ s at the data bus. To carry out this task, the second microcontroller (IC5 at the schematic) will be used. This second microcontroller will be storing all the time inside its internal memory the vibration levels that the other microcontroller (IC4 at the schematic) is sampling, and it will copy the data to an external RAM memory when one of the two situations occurs. While the data is being copied, the input data can be ignored as well as the orders given by the `Store` push button.

To do:

1. Proposal. Pin assignation, interrupts usage and calculations to configure the peripheral elements of each microcontroller (IC4 & IC5). Discussion of what you should be change if two PIC24FJ64GA00X were used. Define the protocol to pass information between the two microcontrollers (`TOSTORE[n-1..0]`) and justify the n value.
2. ADTs diagram—indicate which ADTs include motor- and dictionary of the IC4 microcontroller software.
3. State diagrams of ALL the cooperative motors of the IC4 microcontroller.
4. C code of the initialization of each of the ADTs of the IC4 microcontroller software.
5. C implementation of two of the most significant cooperative motors.
6. Explanation and proposal of the IC5 microcontroller software.
7. Assembly code of the IC5 microcontroller (you do not need to do the initialization).

"The important variables have important names"
Francesc Escudero, master & professor.