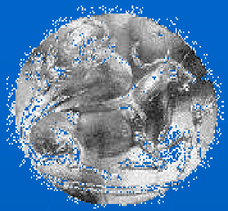
Development of a 28-bit 1.5GHz Frequency Counter



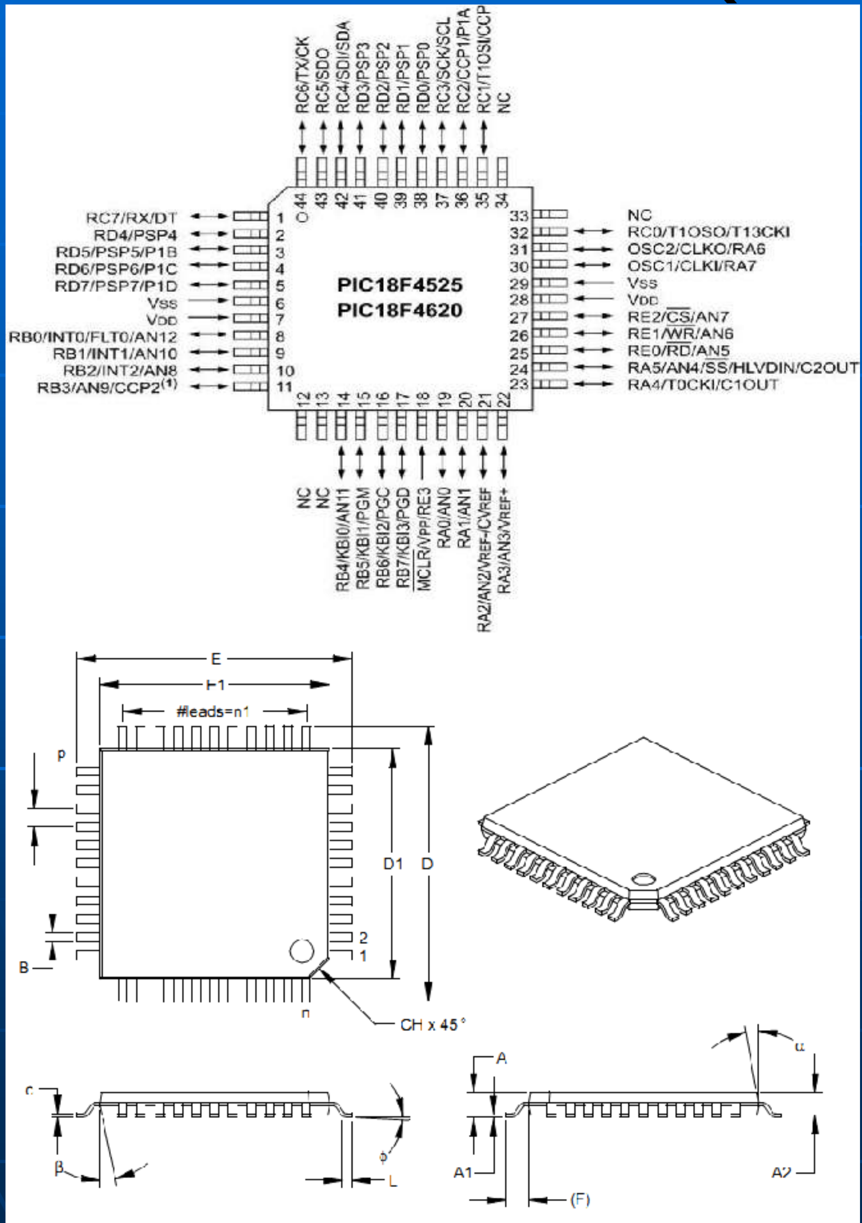
G. Adamidis[1], T. N. Kapetanakis[1], and I. O. Vardiambasis[1]

Technological Educational Institute (TEI) of Crete, Chania Branch, Crete, Greece

[1]Broadband Communications and Electromagnetic Applications (BCEMA) Laboratory,

Department of Electronics

Introduction (1/2)



- Some people say “you’d better build it than buy it”.
- Besides saving money you may also have a lot of fun, acquire new knowledge and experiences.
- This work started as a PIC learning project, since it uses a base 8-bit PIC but it turned out to be more than that.
- Although using an 8-bit PIC, a 28-bit counter is actually implemented.

Introduction 2/2

- The device measures frequency from 0.1 Hz to 1.5 GHz and displays it on a 2x16 character LCD display.
- It offers a frequency resolution up to the 0.1 Hz for frequencies in the range of 0.1 Hz to 100 MHz and ,
- Up to 4 Hz for frequencies in the range of 100 MHz to 1.5 GHz.
- Additionally, Min and Max hold functions, Frequency units selection and, Gate time adjustment are also supported.

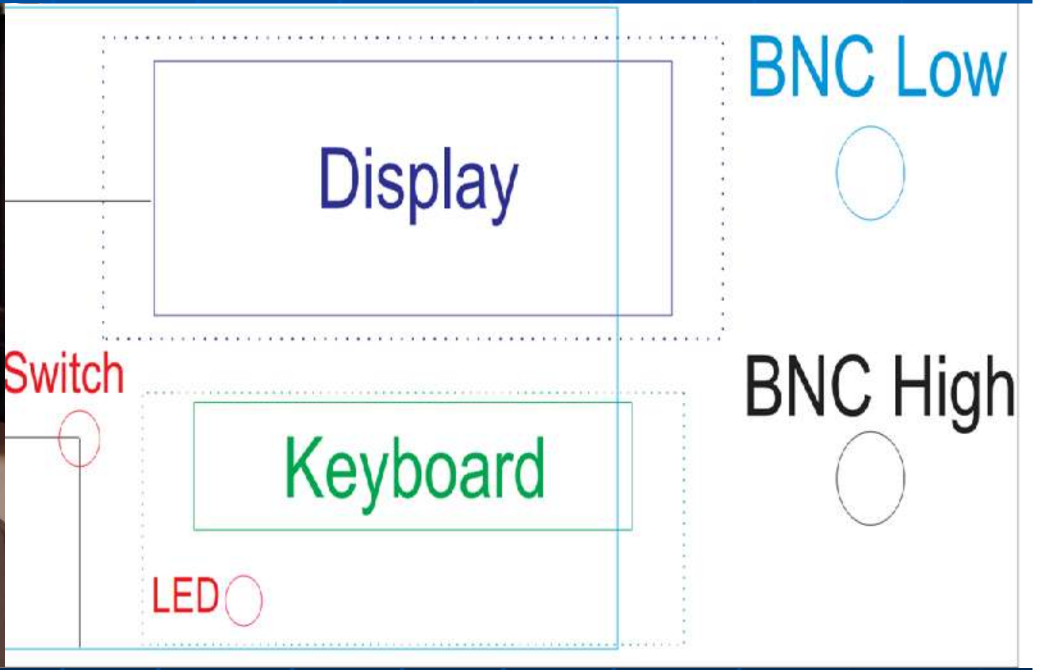


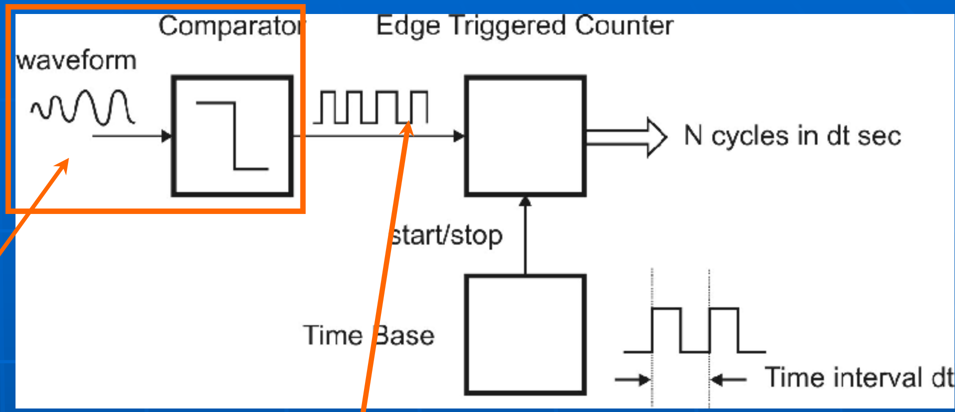
Photo of the proposed device on the left and box mask diagram on the right.

Theory of Operation (1/2)

- The frequency (f) of any periodic waveform can be calculated by counting the instances (N) of the waveform during a precise time interval (dt) from:

$$f = \frac{N}{dt}$$

- The frequency measurement unit is the Hz and 1 Hz is defined as one instance per second. An obvious frequency measuring technique is presented in figure on the right.

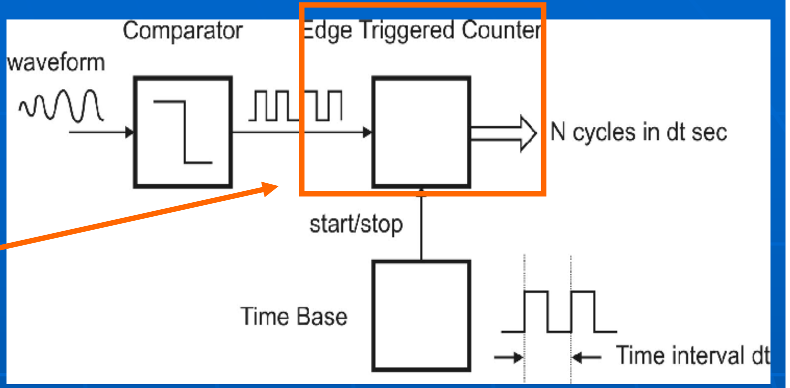


A simple technique for frequency measurement. Consisted from: A Comparator, An Edge Trigger counter and, A Time Base.

- According to this technique, the input waveform must first be converted in an equivalent digital form.

Theory of Operation (2/2)

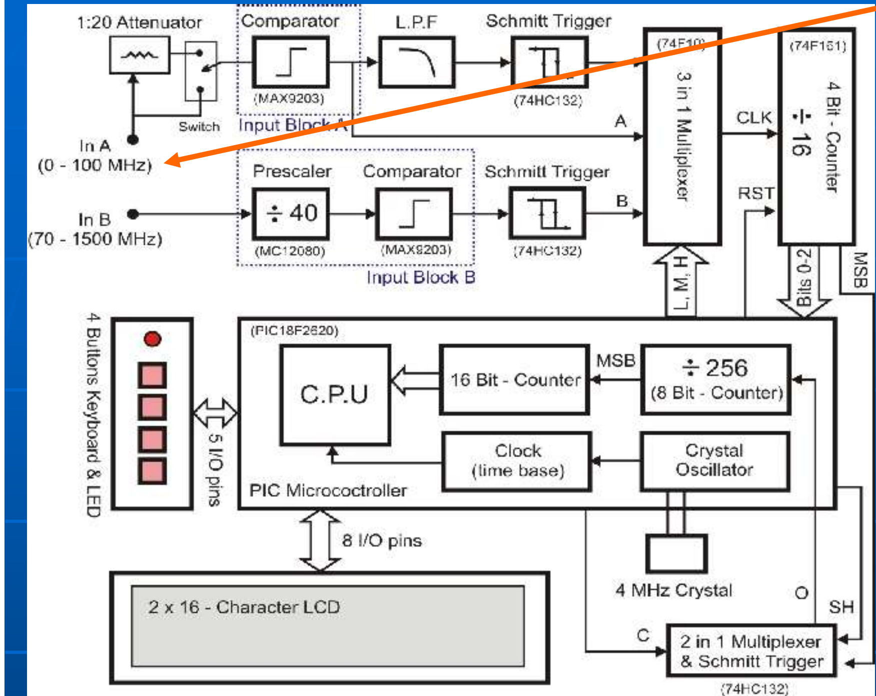
- This digital form is actually a fast switching binary signal which preserves the frequency characteristics of the input waveform.
- Then, an Edge-Triggered Digital Counter is used to accurately count (starting from 0) the N occurred pulses in a precise time interval dt.
- This time interval is provided from an accurate (reference) time base.
- Afterwards, a microcontroller can be used to calculate the measured frequency from f=N / dt and display the result on a common display unit.



A simple technique for frequency measurement. Consisted from: A Comparator, An Edge Trigger counter and, A Time Base.

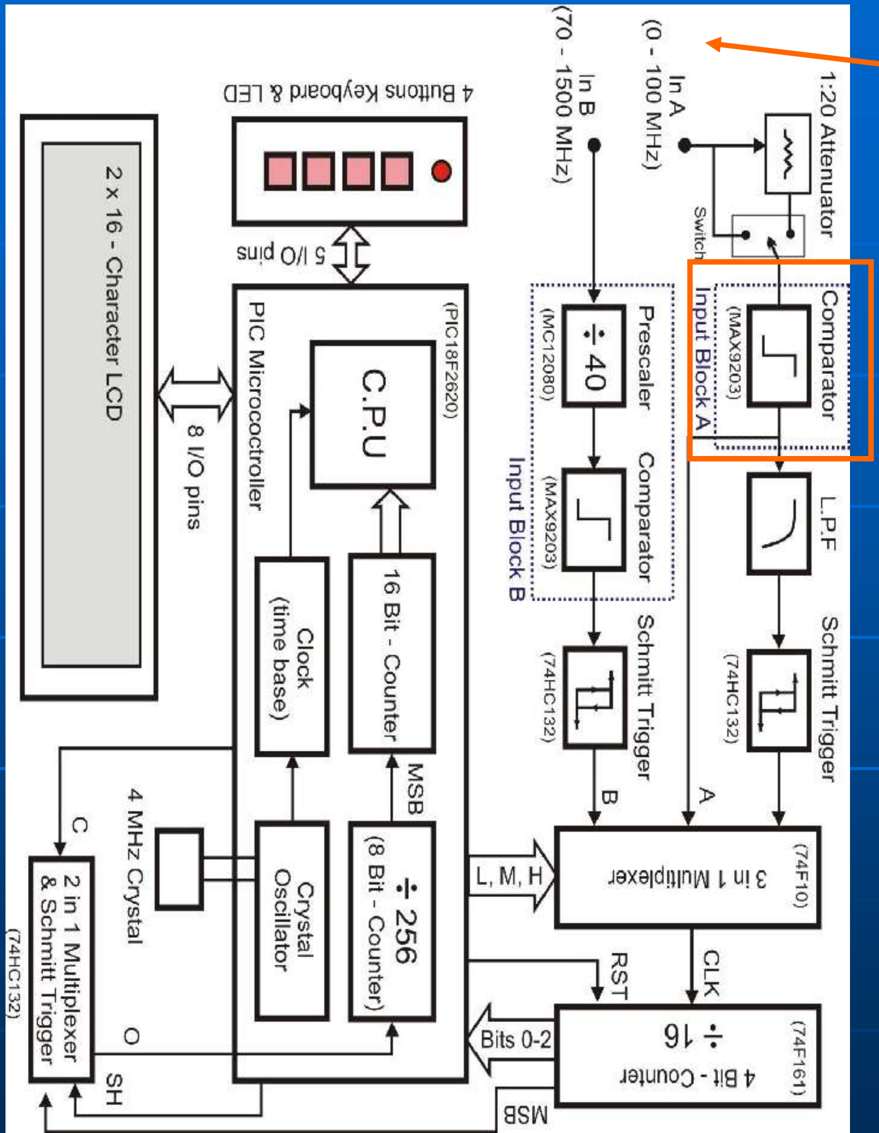
Photo of the implemented frequency counter during test measurement

Hardware (1/6)



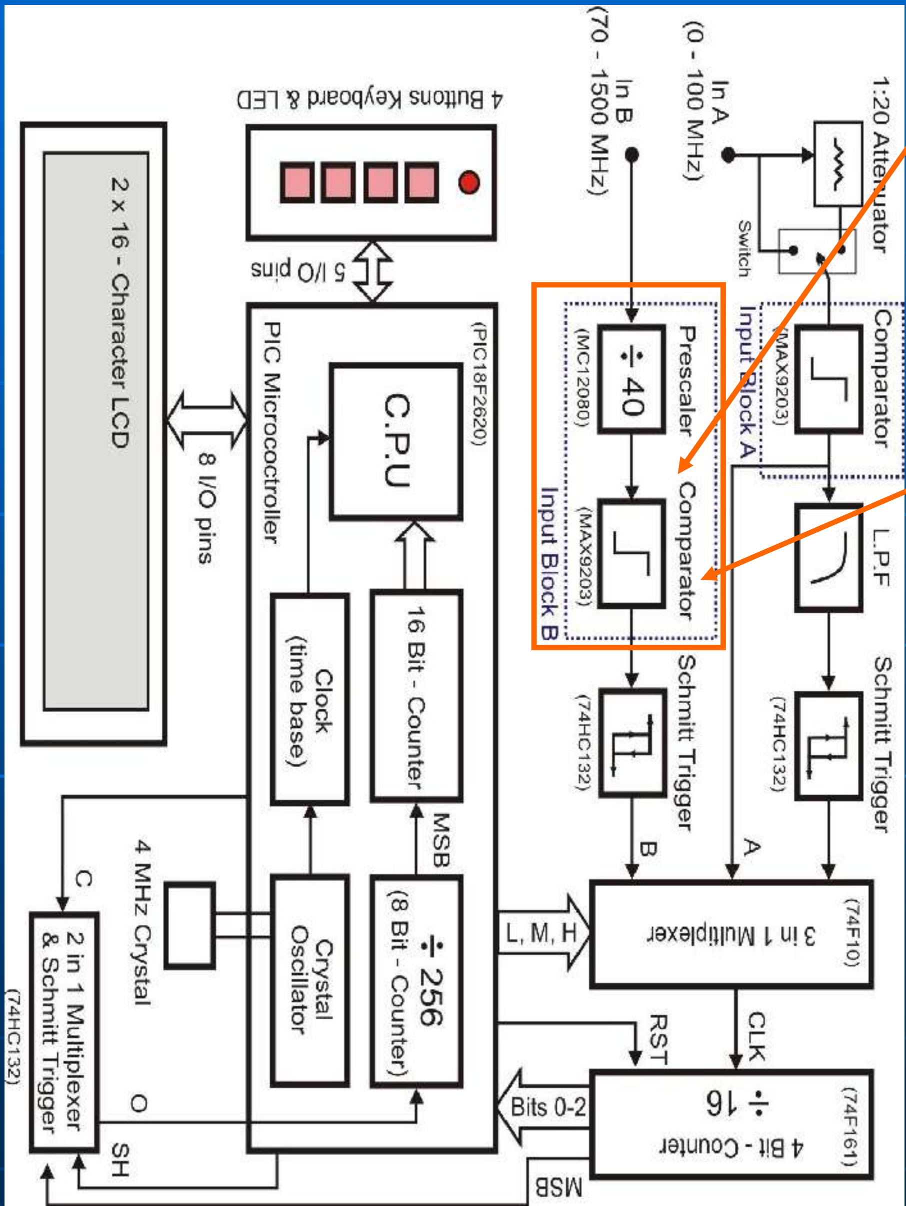
- Each input block has its own input; In A and In B, respectively.
- The first input (In A) covers the 0.1 Hz - 100MHz while the second one (In B) covers the 70-1.5GHz band and the user is able to switch between them by pressing the A / B button.
- The block diagram of the hardware is presented in figure on the right.
- The hardware actually implements the frequency measuring technique of figure on the left.
- Our primary intention was to build a frequency counter with good input sensitivity which would be able to measure signals with amplitude less than some mV at frequencies from almost zero to more than 1GHz.

Hardware (2/6)



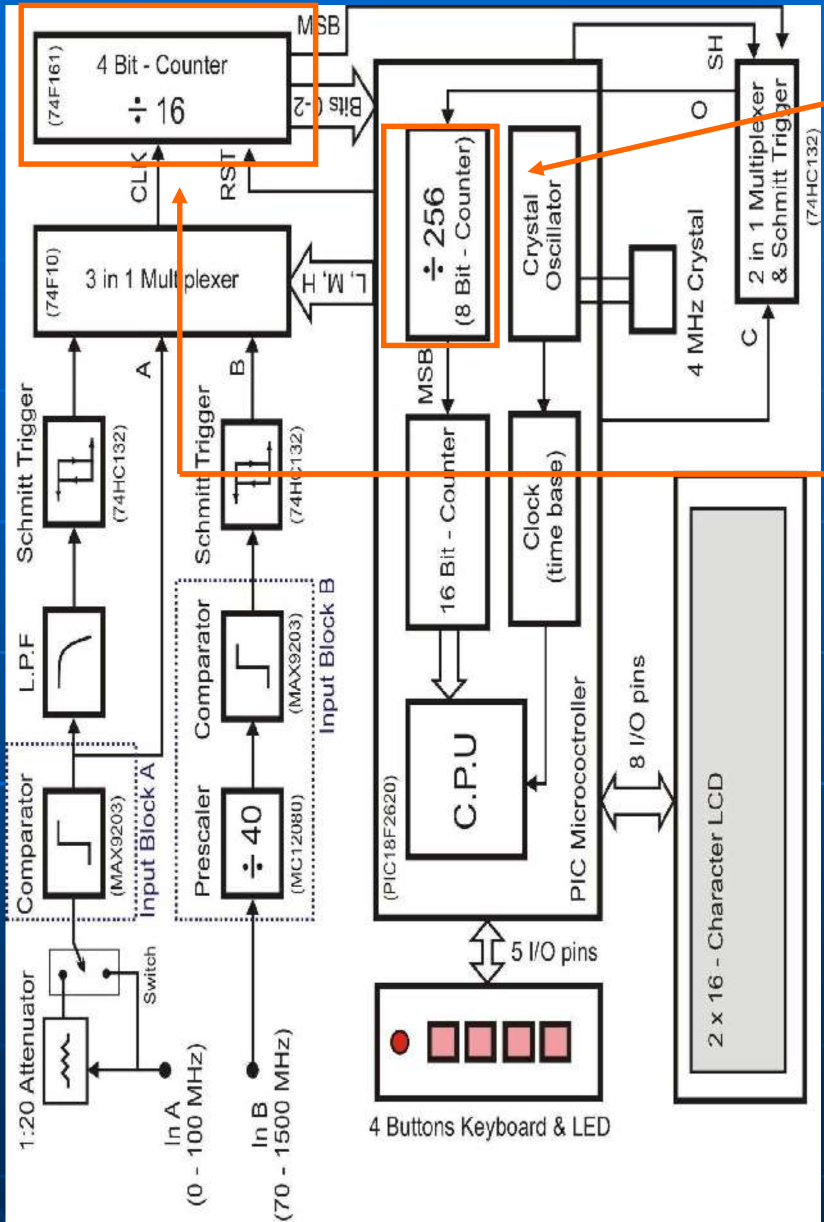
- This wasn’t easy to be achieved with a single input block. So, it was necessary to use 2 different input blocks; A and B input block, respectively.
- The main purpose of the input blocks is to convert the input signal in a fast switching binary equivalent.
- This conversion is better known as Squaring / restoration and it is performed by a MAX9203 high speed comparator in each input block of my design.
- The MAX9203 is a high speed comparator which has a propagation delay of about 7ns and an input sensitivity of about 5mV.

Hardware (3/6)



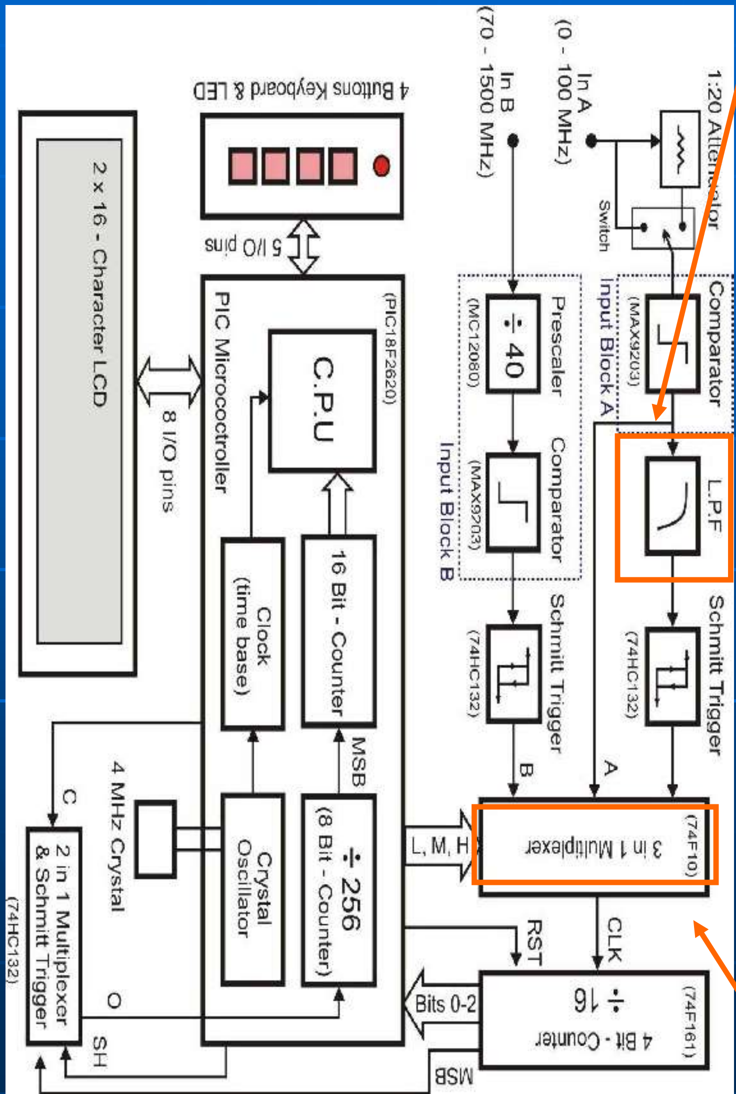
- These characteristics make MAX9203 ideal for frequencies up to 100MHz but an additional prescaler must be used for higher frequencies. This is why we use the MC12080 prescaler in the second input block (B block).
- Using the MC12080, we were able to extend the frequency range of my counter up to 1.5GHz.
- As we mentioned above, my frequency counter is actually a 28bit counter. You maybe wonder how this is possible to be done using a PIC which has only 16-bit internal counters.
- This is because we use the PIC's internal 16-bit Timer0 module (configured as a 16-bit counter), an additional 4-bit external counter (74F161) and the PIC's internal prescaler (in 1:256 prescale mode) in series. That's why I have 28bit in total.

Hardware (4/6)



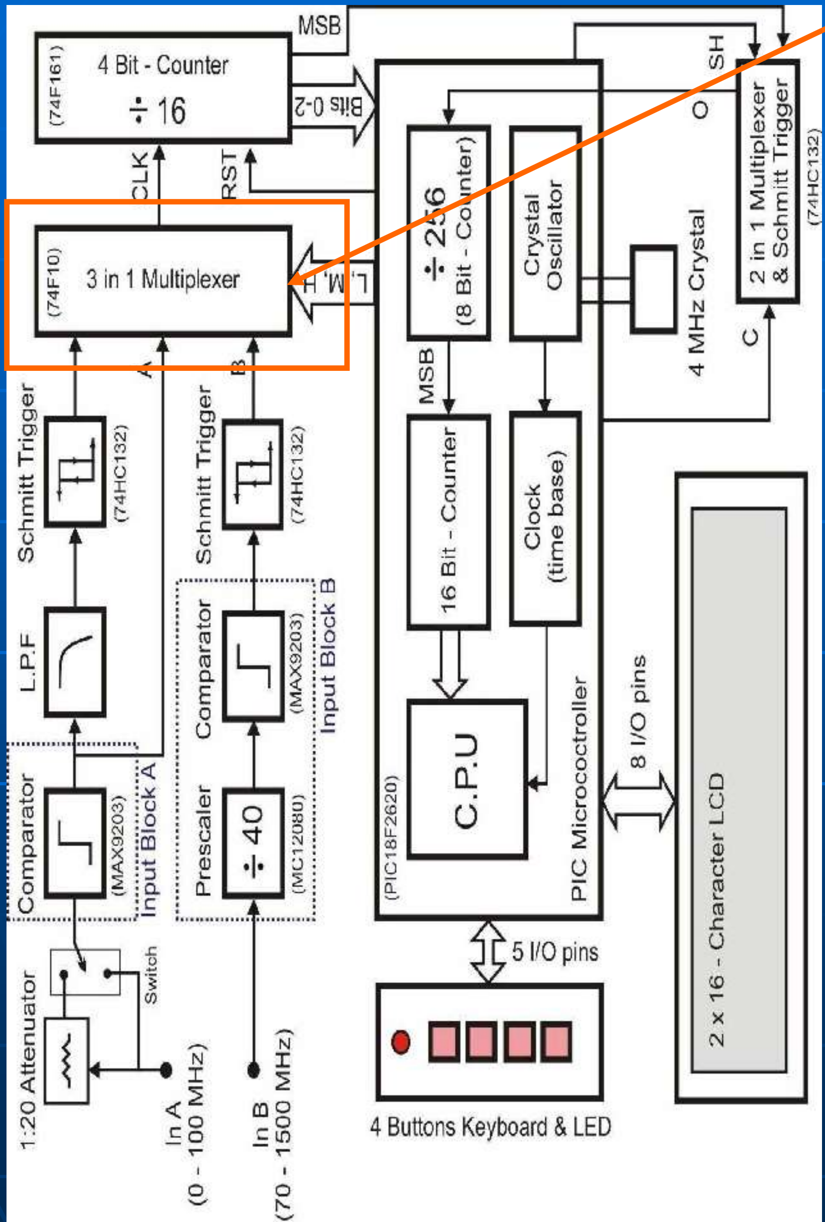
- The PIC’s internal prescaler, in 1:256 prescale mode, is actually an 8-bit counter but it is not directly readable. There is a unique method by which the user can “extract” the 8-bit value of the prescaler which is described in [1].
- By concatenating the calculated value and the original values for timer0 and the external counter (74F161), the 28-bit value for the frequency is determined.
- Rise and fall times of the input frequency in the TOCKI pin are specified to be 10ns, so the fastest clock rate that TOCKI pin can accept is 50 MHZ [1].
- That’s why we expect that a PIC couldn’t be used directly for measuring frequency up to 50 MHz and we maybe wonder how this is overcome in our project.
- This is because we use an external 4-bit counter and the upper frequency limit is determined only by the maximum speed of the external counter and not by the PIC itself.

Hardware (5/6)



- There is another interesting point in this project concerning the LPF (Low Pass Filter).
- As a high speed comparator, the MAX9203 has a high gain-bandwidth product which creates oscillation problems when the input traverses the comparator's linear region.
- These oscillation problems could cause false triggering in the next stage (the counter).
- For clean output switching without oscillation or steps in the output waveform, the input in the MAX9203 must meet typical slew rate requirements of about 0.5V / sec.
- That's why the LPF is necessary in conjunction with a Schmitt trigger to "clean" any oscillation problems in low frequencies.
- Otherwise a false frequency measurement could occur due to any noisy trigger signal in the counter.
- This is also an explanation for the use of the 3 in 1 multiplexer.

Hardware (6/6)



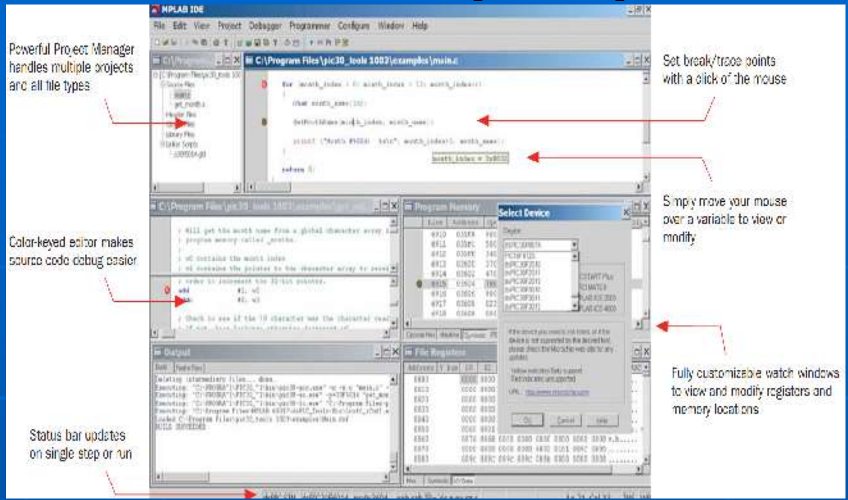
- This multiplexer not only enables switching between A and B input channels but also enables switching to the “cleaned” from oscillations A2 signal at any time this is necessary to be done (at low frequencies) ensuring accurate measurements at low frequencies.
- The input frequency is “gated” for a precise duration of time (dt). This precise “gate” is implemented in software as an accurate delay.
- The “gate” is user selectable from the “mode” button in the keyboard and the user is able to choose between 0.1, 1 and 10 sec “gate”.
- Thus, the expected measurement accuracy is 10, 1 and 0.1 Hz for the In A and 400, 40 and 4Hz for the In B (due to the use of the MC12080 prescaler in 1:40 prescale mode).

Software (1/2)

- PIC was programmed and the code was debugged using Microchip's ICD2 in-circuit debugger & programmer [3].
- The μcontroller runs at an infinitive loop performing in general the following tasks:
  - Step 1: Switch to channel A or B according to current status.
  - Step 2: If channel A is currently selected then decide to use LPF or not *.
  - Step 3: Reset counter.
  - Step 4: Start counter.
  - Step 5: Wait for dt = 0.1, 1 or 10sec (according to current status) for the counter to count. During waiting, check the keyboard periodically. If any button is pressed, then perform the appropriate tasks, update current status and return to step 1- otherwise continue.

**The decicion about the LPF is made as follow: The PIC takes a quick frequency measurement without the LPF (using 0.05 sec gate). If the measured frequency proved to be less than 980 KHz then the PIC switches to the LPF for accurate measurements.*

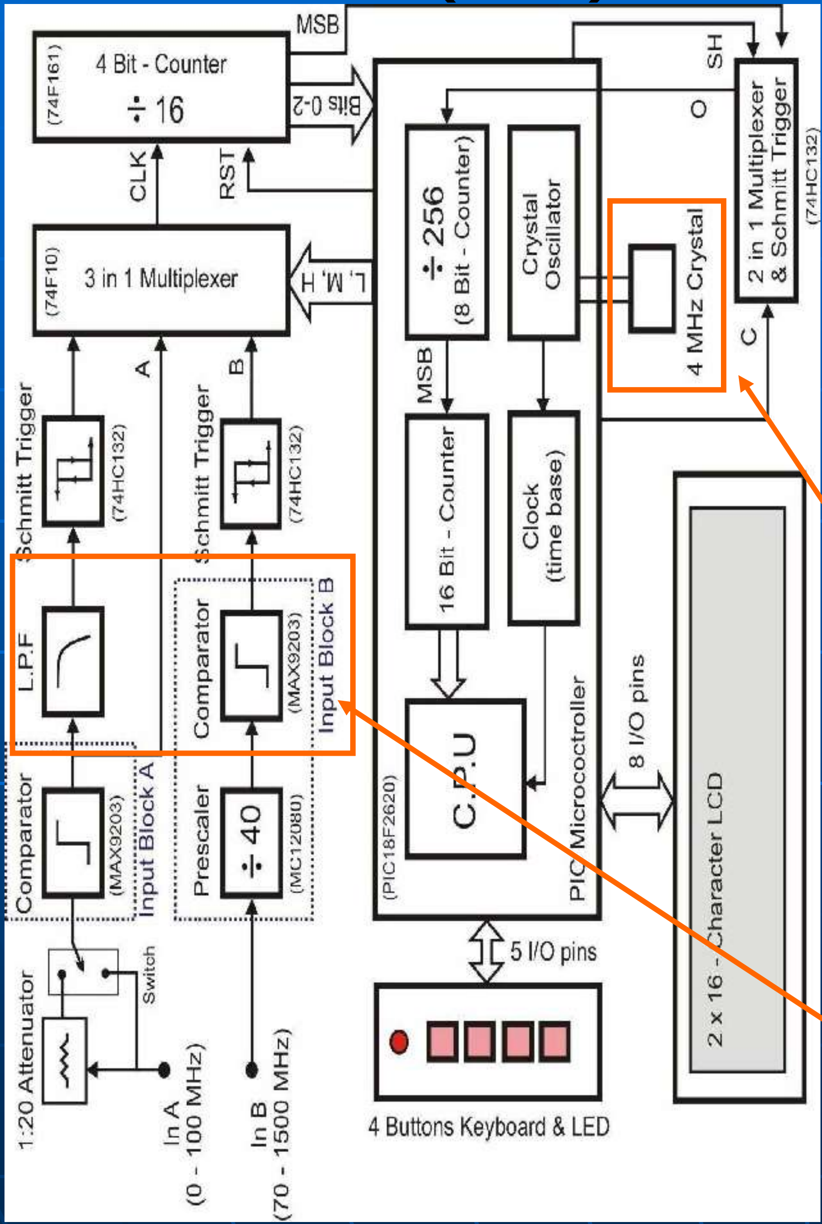
Software (2/2)



- <u>Step 6:</u> Stop counter.
- <u>Step 7:</u> Read counter value N (28 bit integer) and calculate frequency from f=N / dt. **
- <u>Step 8:</u> Display current frequency (real time mode) or the maximum frequency value ever occurred (Max hold mode) or the minimum frequency value ever occurred (Min hold mode) in Hz, KHz or MHz according to current status.
- <u>Step 9:</u> Toggle gate LED.
- <u>Step 10:</u> Return to Step 1.

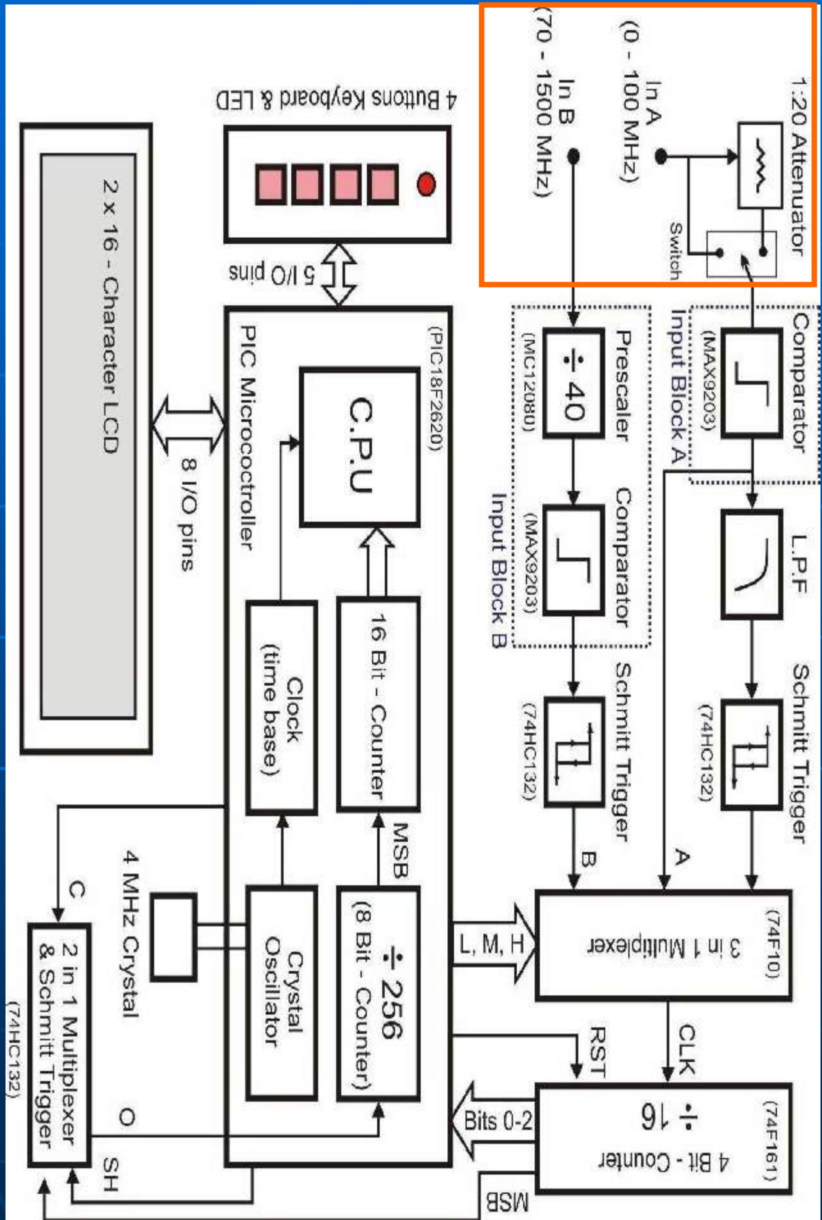
*** The PIC reads the external 4-bit counter's value (E), the 8-bit value of the internal prescaler (P) and the 16-bit value of the internal Timer0 (T) and calculates the 28-bit counter value (N) from: N=E+16•P+4096•T. If the currently selected channel is A then the measured frequency f is simply f=N/dt, but if the current selected channel is B then the measured frequency is calculated from f=N•40/dt (taking into account the MC12080 prescaler which is configured for 1:40 prescale).*

Results (1/2)



- During measurements we noticed that the frequency accuracy was not as expected for frequencies up to 50MHz.
- There was about 800Hz difference between the measured frequency and the reference frequency (from the IFR reference generator) at 1.5GHz and about 60 Hz at 100 MHz (with 1sec gate – at room temperature).
- Although, it seems logic for the 5ppm reference crystal used in this project and with the absence of temperature compensation.
- As about the input sensitivity, the results are presented in the table below.
- As it can be seen, there is a degradation in input sensitivity for the low frequencies.
- Which is possible due to the LPF and to the performance of MAX9203 in low frequencies. 0

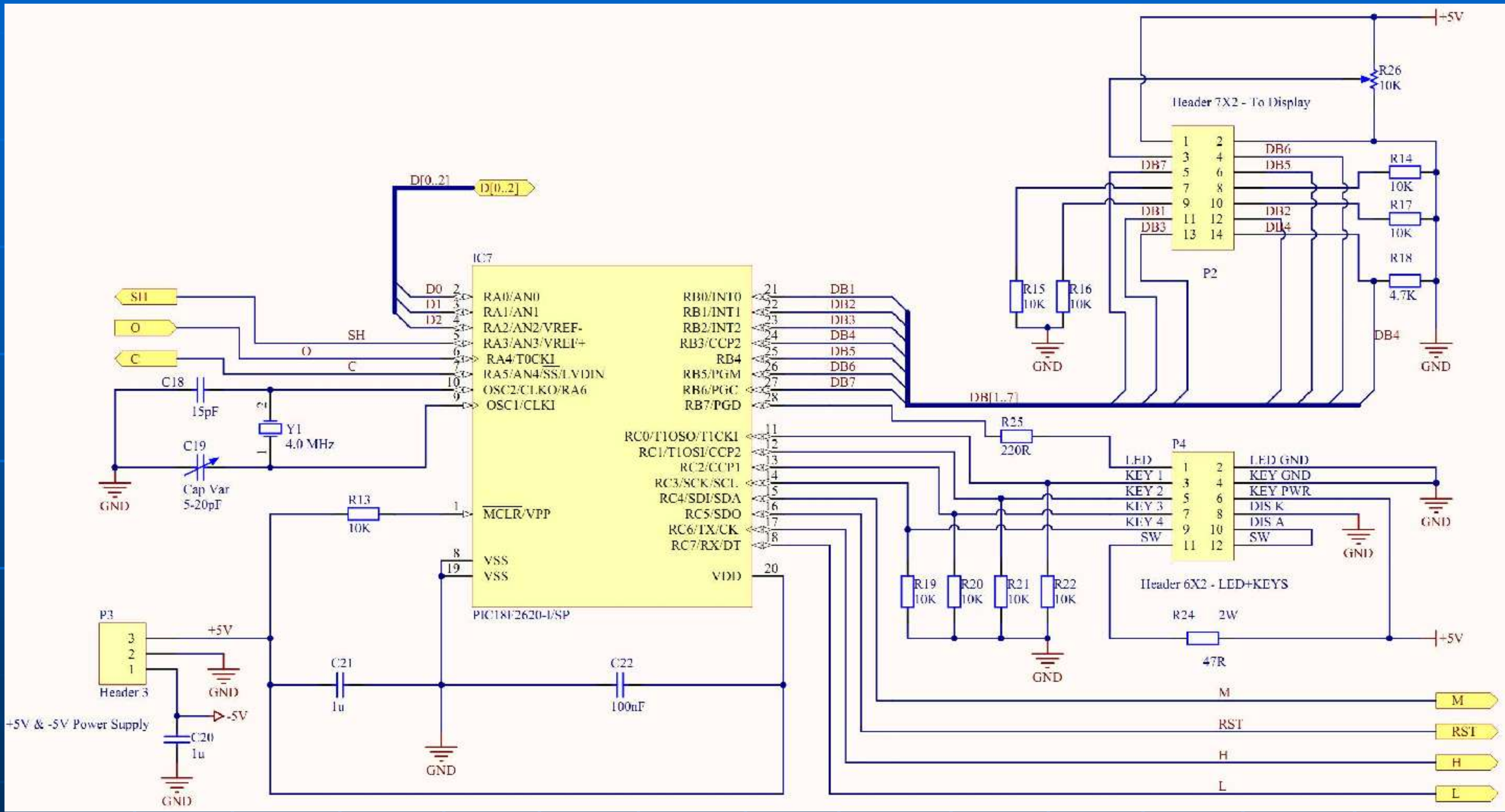
Results (2/2)



- Should be mentioned that the maximum signal amplitude which can safely be applied in the In A is about 10Vp-p without the use of the attenuator and about 200Vp-p with the use of the 1:20 attenuator.
- As about the In B, the maximum signal strength must not exceed +13dbm.

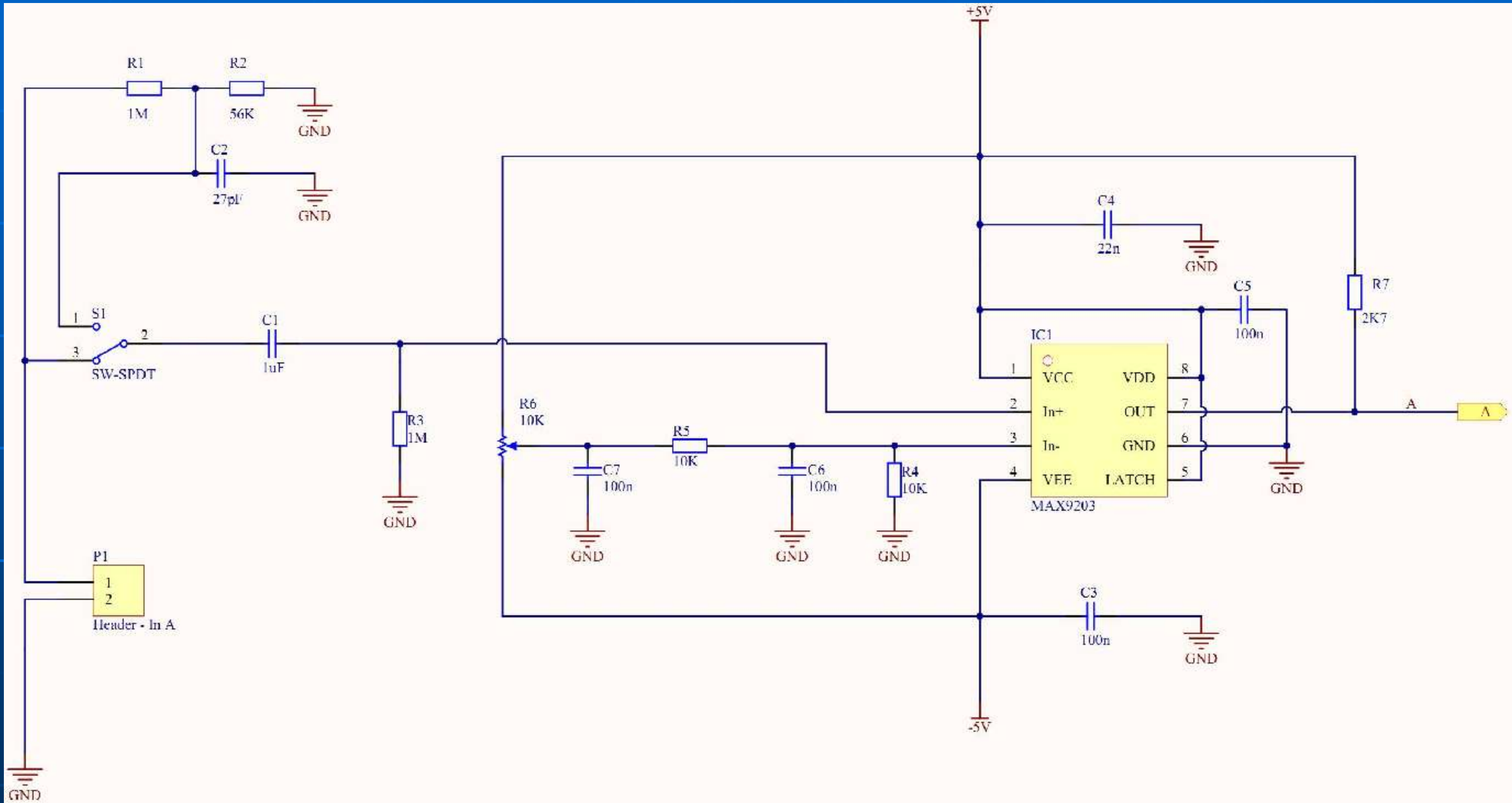
Frequency	In A sensitivity (mVp-p)	In B sensitivity (dbm)
10 Hz	120	——
1 KHz	60	——
100 KHz	20	——
1 MHz	10	——
70 MHz	10	-14
200 MHz	——	-19
500 MHz	——	-20
1 GHz	——	-18
1.5 GHz	——	-7

Schematics (1/2)



μ-controller wiring.

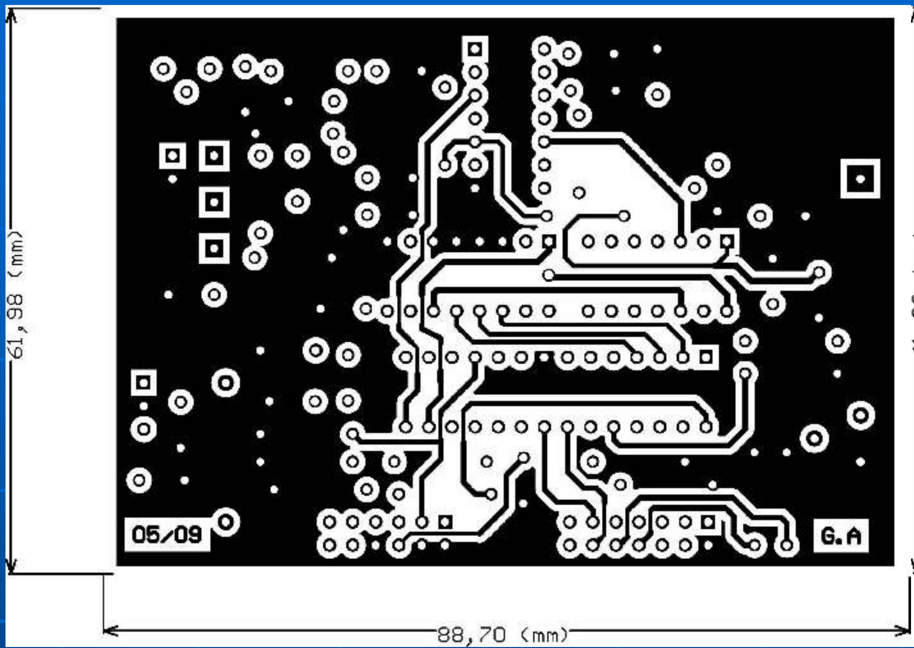
Schematics (2/2)



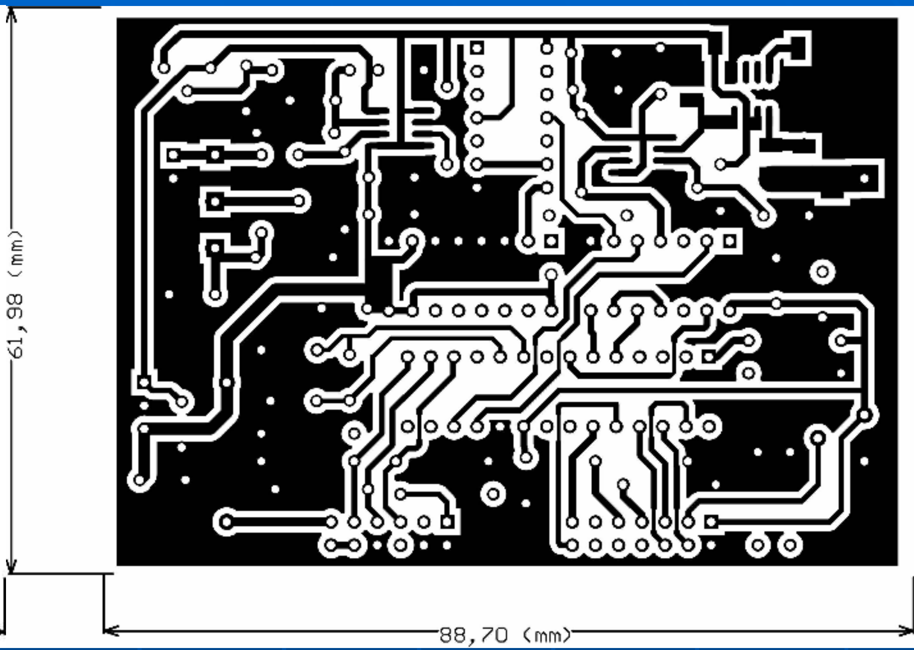
High speed comparator (MAX9203) wiring

Composite Diagram-Printed Circuit Board

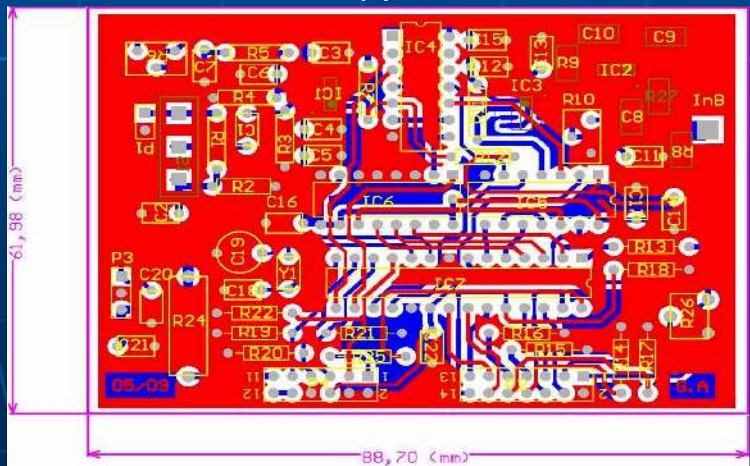
(a)



(b)



(c)



(a) Top Layer, (b) Bottom Layer of the printed circuit board.

(c) Composite diagram of the proposed frequency counter

Further Directions

- We will make some improvements in the near future, regarding the temperature compensation in the reference clock and regarding the input stage. (Possible solution: a NTC resistor as a heating element in order to keep the reference crystal in an almost stable temperature)
- We will probably use a stable temperature heating source for the reference crystal and,
- We will add a section in the A input block in order to improve the input sensitivity in low frequencies.

References

[1] Microchip, Frequency Counter Using PIC16C5X, Application notes AN592, DS00592D, Microchip Technology Inc. 1997.

[2] Microchip, MPLAB C18 C Compiler Libraries, DS51297D, Microchip Technology Inc. 2004.

[3] MPLAB®ICD 2 In-Circuit Debugger/Programmer [4]Microchip, PIC18F2525/2620/4525/4620 Data Sheet, DS39626B, Microchip Technology Inc. 2004.

[5] MAXIM, MAX9201/MAX9202/MAX9203 Data Sheet, 19-1936; Rev1; 1/05.

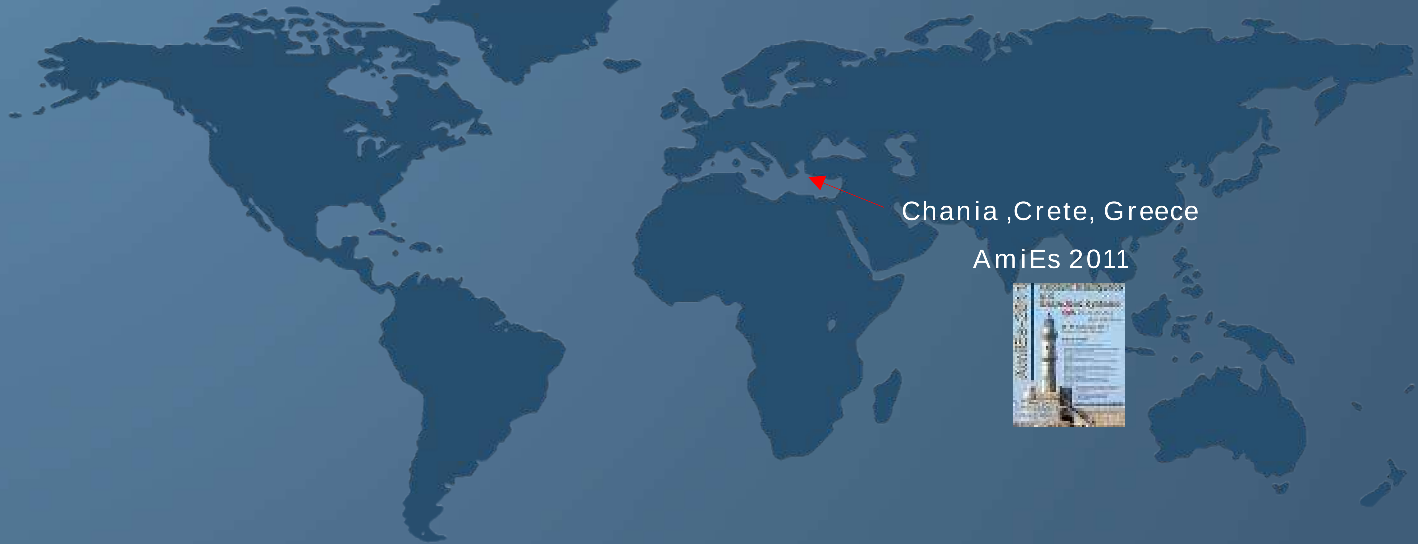
[6] Freescale Semiconductor, Inc, MC12080 Data Sheet, Motorola, Inc. 1999 Rev 3.

[7] Philips Semiconductors, 74HC/HCT132 Quad 2-input NAND Schmitt trigger Data Sheet, September '93.

[8] Philips Semiconductors, 74F161A / 74F163A 4-bit binary counter Data Sheet, 1996 Jan 29.

Development of a 28-bit 1.5GHz Frequency Counter

G. Adamidis, T. N. Kapetanakis, and I. O. Vardiambasis



Thank you !!!

Broadband Communications & Electromagnetic Applications (BCEMA) Laboratory,
Department of Electronics
Technological Educational Institute (TEI) of Crete,
Chania Branch, Crete, Greece