FREQUENCY METER

Omar Yasser, Omar Hosni, Omar Ahmed, Omar Mohamed

Faculty of Engineering

Course: Embedded Systems

Course Code: EEC 341

Dr. Hossam Eldin Mostafa

12/23

Overview

- This is probably the simplest frequency counter on an AVR microcontroller. It allows you to measure frequencies up to 10 MHz in 4 automatically selected ranges. The lowest range has a resolution of 1 Hz. A 4-digit LED display is used to display the measured frequency. The device is based on the Atmel AVR ATtiny3413, ATtiny2313A or ATtiny2313 microcontroller.

-The microcontroller is clocked from a quartz resonator with a frequency of 20 MHz (the maximum allowable clock frequency). The accuracy of the measurement is determined by the accuracy of this crystal. The minimum half-cycle length of the measured signal must be greater than the frequency period of the crystal oscillator (MCU architecture limitation). Thus, at 50% duty cycle, frequencies up to 10 MHz can be measured.

-The measured signal is supplied to pin PD5(T1). Counting is performed using a 16-bit timer/counter1, which is synchronized with an external signal. A timer overflow increments an 8-bit register, resulting in a 24-bit frequency value. This value is then converted to decimal form and displayed on the display. The frequency is always displayed in kHz. Automatic range selection changes the position of the decimal point. Refresh rate 1 Hz.

-The indicator cathodes are connected to port B, the anodes to pins 0 - 3 of port D. The super-bright display eliminates the need to use transistors to amplify the anode current. The display is controlled by multiplexing and connected in the usual way. The multiplexing frequency is 156.25 Hz. You can use a display such as CA56-12SRWA. Resistors R1 - R8 determine the current consumption of the display and, therefore, its brightness. They are selected so that the current does not exceed the pin's maximum output current (40 mA).

-This frequency meter is powered by a 5V (+/- 10%) power supply. Consumption at 5 V is about 15-35 mA, depending on the number of illuminated segments (the LED indicator has the highest current consumption).

-If the meter input is in the air, the meter may display meaningless values because the input impedance is high. You can prevent this by connecting a resistor of about 100k ohms between the input and ground.

-Frequency ranges:

Range 1 ... max. 9.999 kHz, resolution up to 1 Hz.

Range 2 ... max. 99.99 kHz, resolution up to 10 Hz.

Range 3...Max. 999.9 kHz, resolution up to 100 Hz.

Range 4...max. 9999 kHz,

Hardware Requirements

1- ATtiny 4313
2- 4digit 7-Segment (Common Anode)
3- 20MHz Crystal Oscillator
4- 8 Resistors 560 Ω
5- 1 Resistor 270 Ω
6- 1 Resistor 100 KΩ
7- 1 non polarized capacitor 12 pF
8- 1 non polarized capacitor 22 pF
9- 1 non polarized capacitor 100 nF
10- 1 polarized capacitor 470 uF, 10 V

Software Requirements

1- Microchip Studio
2- AVRDUDESS

Steps

1- Define Requirements:
   - Determine the frequency range from 1HZ to 10MHZ.
   - Decide on the accuracy and resolution.

2- Choose a microcontroller:
   - We choose the ATtiny 4313 due to its similarity to ATtiny 2313.

3- Connect the crystal oscillator:
   - Connect a crystal oscillator 20MHZ to deal with a very high change in frequency input.

4- Input Signal Conditioning:
   - We use a resistor 220 ohm to limit the input to prevent break down of microcontroller and put resistor 100 k ohm to prevent the floating number in case of no input.

5- Connect the display:
   - We use a 4 digit 7-seqment due to a desired resolution and it is a cheap option.

6- Testing:

- We tested our project on Arduino which we write a code to generate frequency but that is not enough to make sure that our project does very well, we use function generator for more accuracy.

Problems

We faced a few problems which we can talk about at a specific point.

1- Select the microcontroller:

- Due to the original code was written to ATtiny 2313 and this microcontroller is not available, that creates a problem.
- Finally, after searching we found that ATtiny 4313 has a big similarity to ATtiny 2313.

2- Compile the code:

- Because we are changing the microcontroller, we must compile the code to suit the new microcontroller and that done by Atmel studio and change the include library for 4313.

3- Flash the code:

- We used to flash the code USBASP, but the problem is centered at the program to flash the code which most of them do not include the ATtiny 4313.
- Finally, we found AVRDUDES that contain ATtiny 4313.

Testing

- We testes our project by two ways.
  1- Arduino:
  We write the code to generate frequency from 12 HZ to 8 MHZ
  2- Function generator:
  We use the function generator to make sure that the project works very well with different signals.

Limitation

- It cannot read frequencies higher than 10 MHz.

Conclusion

- - Finally, the project is working very well with different signals and ranges without any problem.
- All this was done under the supervision of Dr. Hossam.

Reference

- For Arduino code:
    - https://youtu.be/wk_pxRhVNgA?si=CRpHysGsJoBhft0u
- For frequency counter:
    - https://radioparty.ru/device/avr/686-frequency-i-meter-attiny2313