Lourdes Alicia Garcia Peña - A01551772

Jorge Antonio Vizcarra Vega - A01630448

Julio César Almada Fuerte - A00226046

Career: ITE

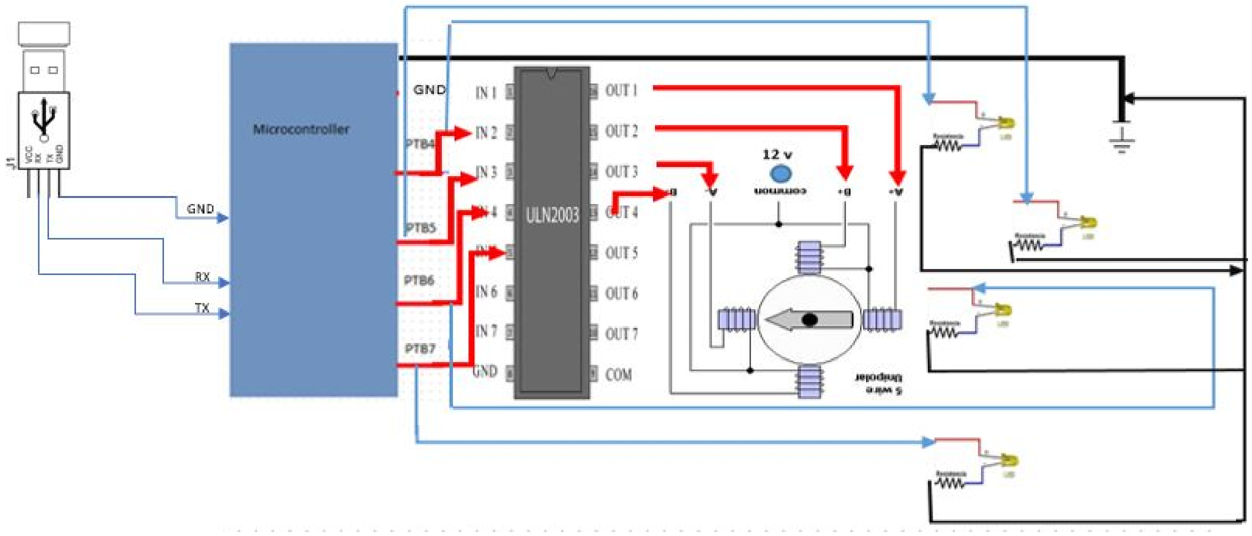
Practice 4: Serial Port

Description:

This practice is about moving a stepper motor being controlled by serial communication and a temperature sensor(controlled by a potentiometer).

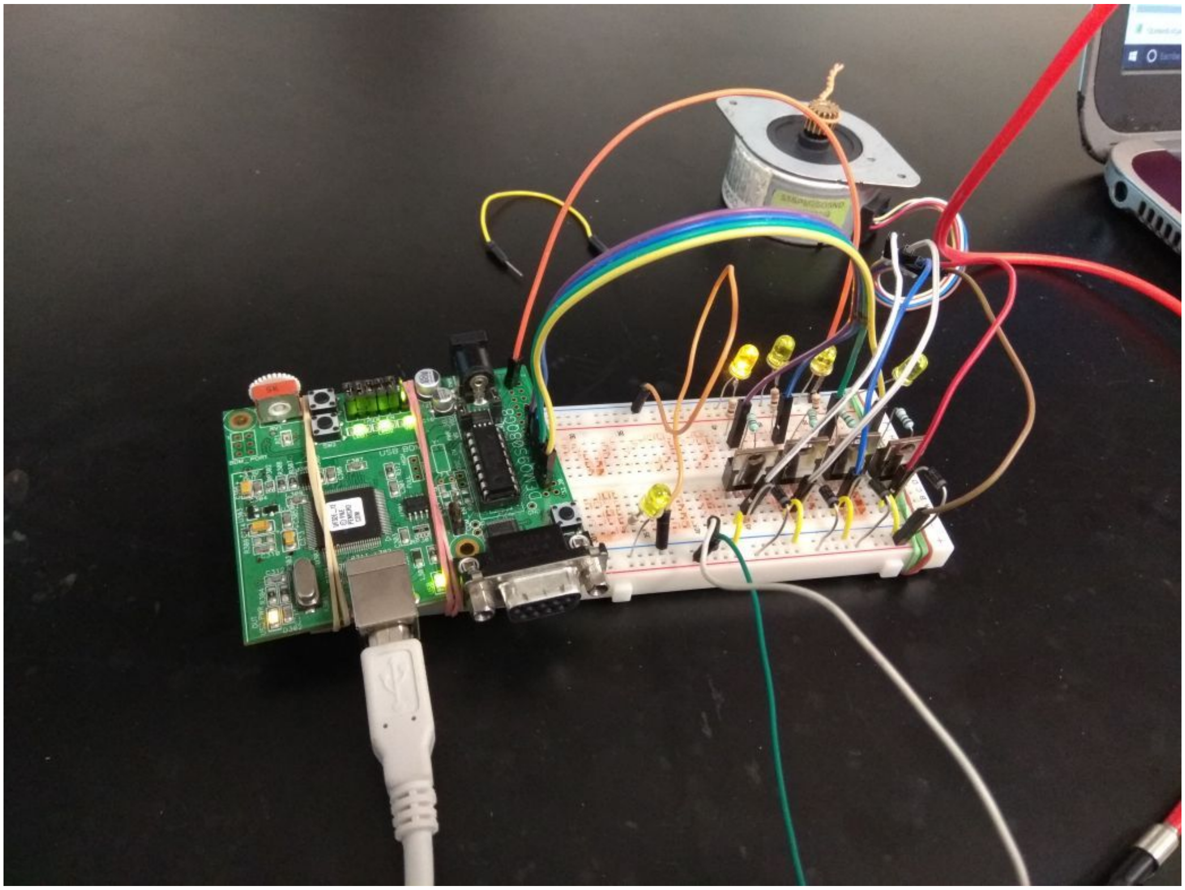
The motor receives various commands to control its velocity, the direction in which it turns, on/off commands , to move it in a certain degree, receive a temperature limit, send messages when receiving a command and when the temperature limit is met.

Connections



Circuit built for this practice

For this practice we used the same hardware that we used for practice 2 and 3, but adding the TTL usb in order to allow serial communication.



Decisions taken:

For checking each command we used a switch case, so it does not need to enter into every “if” clause.

We did not process in the interruptions, instead, we set flags that enabled each method.

The layers are an important part of the code because with them we made the code clearer and easier to read, but above all to manage. The layers we used are

- RXTX : to allow serial communication, receiving and transmitting messages in ASCII.
- Validation: In this layer there are methods used to translate from ASCII to binary and the opposite.
- MotorDriver: To control every step of the motor.

- ADC: Interruptions of ADC module are here and methods to store the values of it.
- GPIO: This layer is where every port is initialized.

Problems met during the development of the Project:

One of the problems we met was making the serial communication work, because we didn't really see how it worked with Sinsel, well we just saw a simple example of how it worked. Leaving the transmission on gave us a lot of trouble because it caused many glitches, and other parts of the code didn't worked properly because of this problem. It was what took more work, everything else was just pure simple programming and finding the formulas for everything else.

Conclusion:

Julio:

I liked this lab practica, it certainly was very hard to make the serial communication to work, but with some help we understood how everything work, also making tests and debugging. what concerned me is how the debugger works in a strange manner from time to time, and logic seems good but when it comes to checking it, something is wrong and we had to check code all the time. When enabling the transmission it created many problems for other parts in the code so we had to change a lot of code.

Lourdes:

Until now this has been the most difficult practice because it was very long. There are many functions on it and not that much time to make it happen successfully. The serial communication was the part which took more time, we needed many flags to make it possible, otherwise it was transmitting what the buffer had and I think there should be a register in the microcontroller that allow us to clear the buffer just like you can turn on and off a module.

Jorge:

This is the most difficult practice we have done because it implied many functions in our code and this took a lot time to develop it. The main difficult was the SCID buffer because this buffer works for transmission and reception so when we want to transmit or receive information we have to turn on or turn off many flags in order to have the control and decide what function is the correct to use. At the end we have to waste a lot of time debugging our code because we had many problems with the transmission.