

Week 1 report

Javier Torres

January 3-5, 2017

Contents

1	Advance Status	1
2	Comments and Thoughts	1
3	Next Steps	2

1 Advance Status

Programming the LPC2148 I was able to burn a example code (a simple “blink” program) in the LPC2148, from now on referred as “the MCU”. This was done using ISP via on-chip boot-loader through UART0 port of the MCU and an USB to Serial converter (FTDI FT232RL chip) between the MCU and the computer.

Reading data from LiDAR Lite V2 I wrote a library for the “LiDAR Lite V2”, from now on referred as “the sensor”, to work with the MCU via I2C communication. This was done according to the “Operational overview”, “I2C protocol summary” and “Detailed register definitions” sections of the “LiDAR Lite Documentation” file. The library has all the functions needed to configure and read from the sensor. The library still need to be tested.

2 Comments and Thoughts

About the information sources As I had no prior experience on this particular MCU and sensor, a lot of reading was needed. The main sources of information were “Lidar Lite Documentation” from the official GitHub repository of “Pulsed Light 3D”, “LPC214x User Manual” and “LPC2141/42/44/46/48 Datasheet”. The latter was obtained from the official MCU product page at the “NXP” website and “LPC2148 board header assignment and schematics” from the official dev board page at “Olimex” website.

About the code uploading process The main problem that I had to solve was to find a way to burn code to the MCU flash memory. At the beginning I tried with the on board JTAG connector, as it's the method I knew, but I couldn't find a JTAG interface. After a while, I finally found one. Unfortunately, it didn't come with a license for its companion software, so I couldn't proceed until I got a license or found another compatible software that didn't asked for a license. Finally I decided to look for other options. The solution was an USB to Serial converter and the "Flash Magic" software using ISP via on-chip boot-loader through UART0 port of the MCU. The USB to Serial converter used is a "FTDI Basic" board from Sparkfun. Boris Vidal, who was working in the same laboratory during Thursday 05/01 on another project with professor Auat had a spare one.

About the measurement rate Many aspects of the sensor operation can be configured through the library. If set properly, it can even reach a 750Hz maximum measurement rate (versus the 100Hz by default) at the expense of somewhat reduced sensitivity and maximum range.

3 Next Steps

Sensor library testing In order to test the sensor library I will use the USB to serial converter and UART0 port of the MCU to communicate between the MCU and computer. On the computer side, I will have a simple C program that reads information from USB to Serial converter port and displays it on the terminal. The MCU will be reading the sensor through I2C and sending the measured distance to the USB to Serial converter through its UART0 port.

Testing the sensor Once the library is working properly, I will proceed to test the measurement accuracy for different sensor configurations, in particular to achieve sensor measurement rates above the 100Hz default rate.

Stepper motor Finally, I will start working on the code to control the stepper motor with the provided motor controller.