MIPS32 Processor on DE0-Nano with Peripheral Shields

SPONSOR:
IEEE 5001 STANDARDS

SPONSOR ADVISOR:
ERIC LINDBERG

FACULTY ADVISOR:
NICOLE HAMILTON

TEAM MEMBERS:
DANIEL LUNCASU-ROLEA
TREYVEN CHIN
JOSHUA YANG
RENQING LI

The Team

Daniel Luncasu-Rolea
- Hardware Design
- Programming
- SMD Soldering
- PCB Revision
- Presentation/Poster

Joshua Yang
- Bottom PCB Design
- Code testing
- Inventory

Treyven Chin
- Top PCB Design
- Inventory

Renqing Li
- Programming
- Presentation
- Poster

Our Product

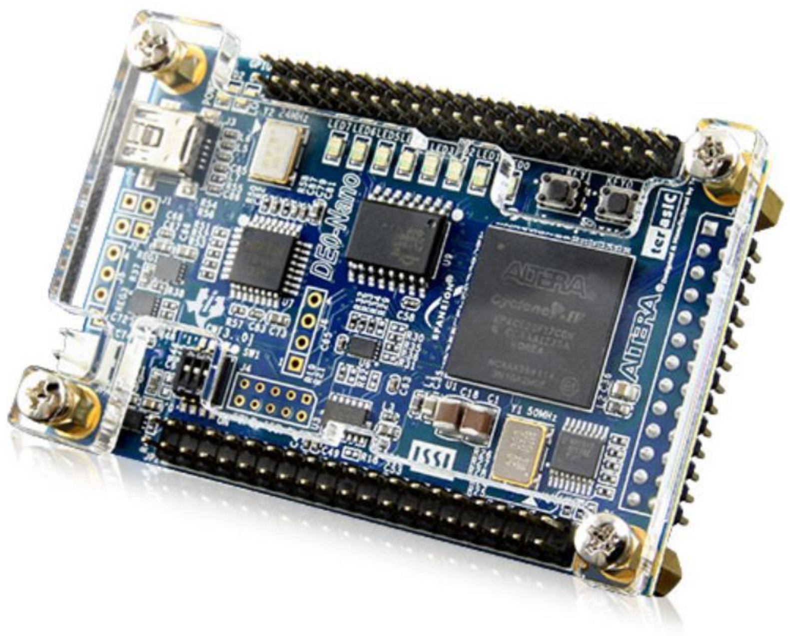
A softcore microcontroller on a DE0-Nano FPGA.
•Useful for students and hobbyists

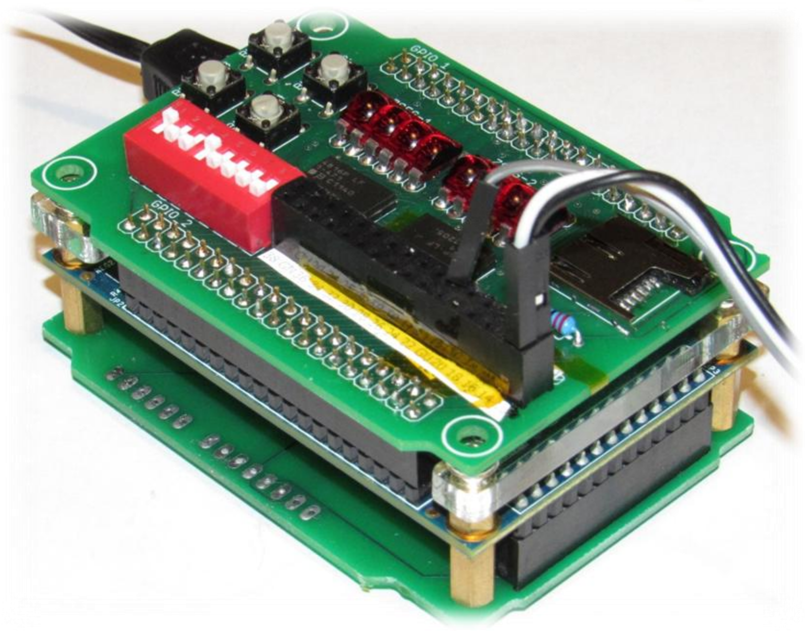
Key Features
•32-bit, 50 MHz RISC processor
•32 MB SDRAM chip
•GCC cross-compiler for C
•Top board with I/O Devices

- 7-Segment displays
- Buttons
- Switches
- MicroSD Socket

•Bottom board with Arduino headers

- 3.3V to 5V step-up
- 3.3V to 5V logic level shifter





DE0-Nano FPGA (Field-Programmable Gate Array)

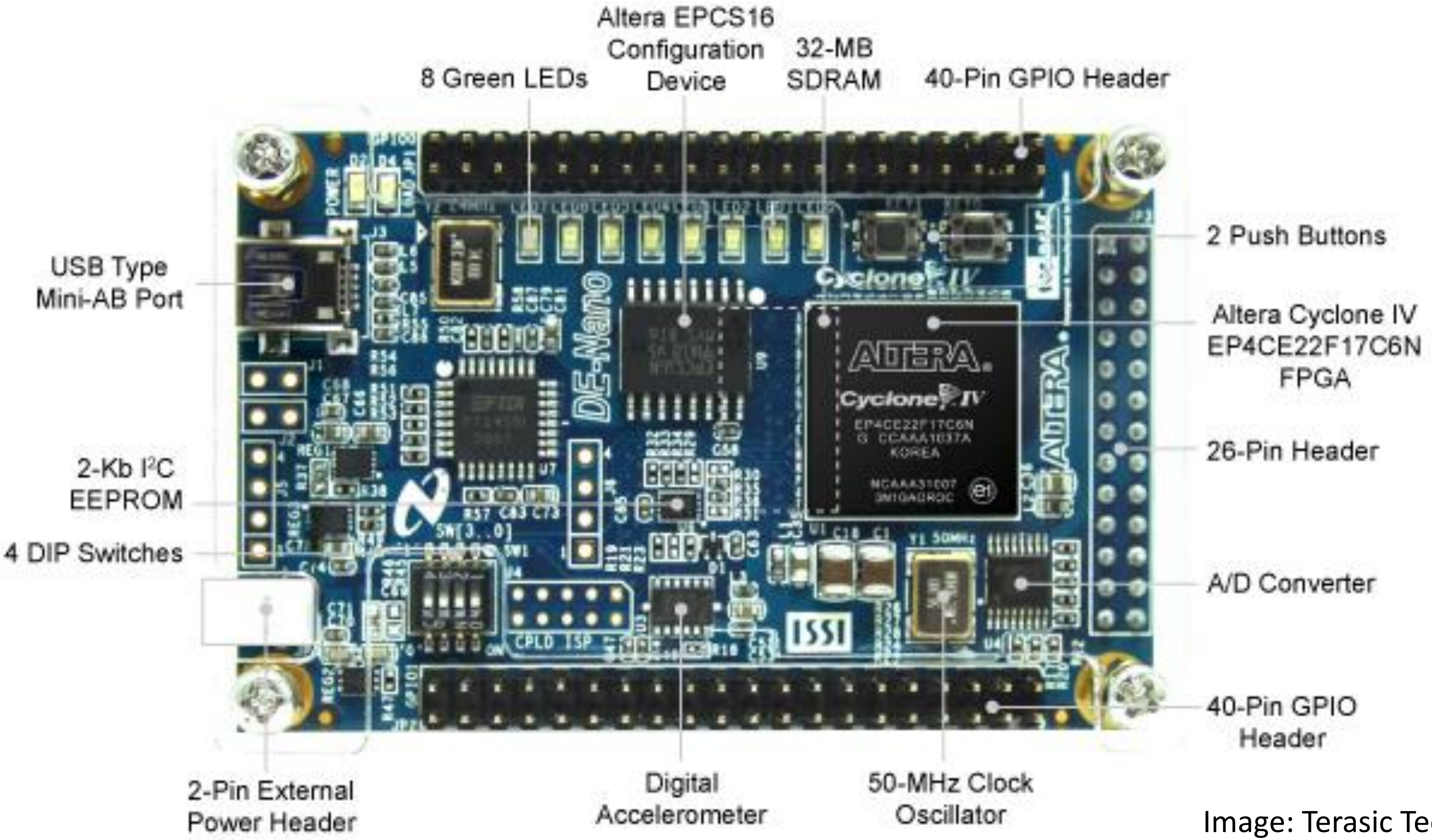
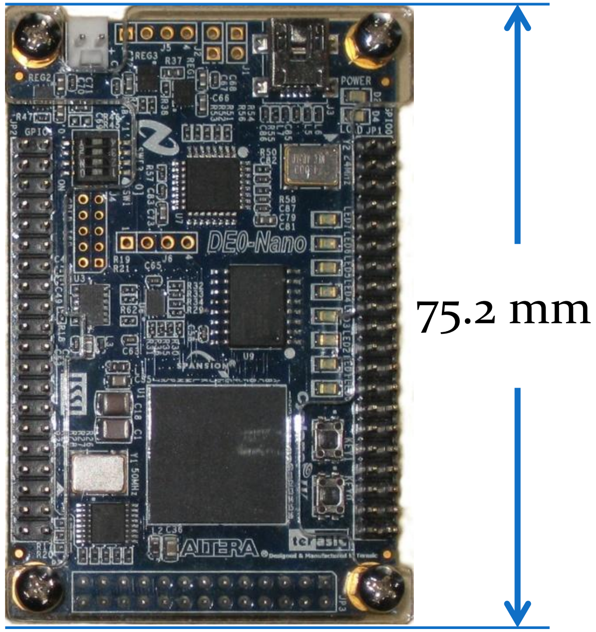
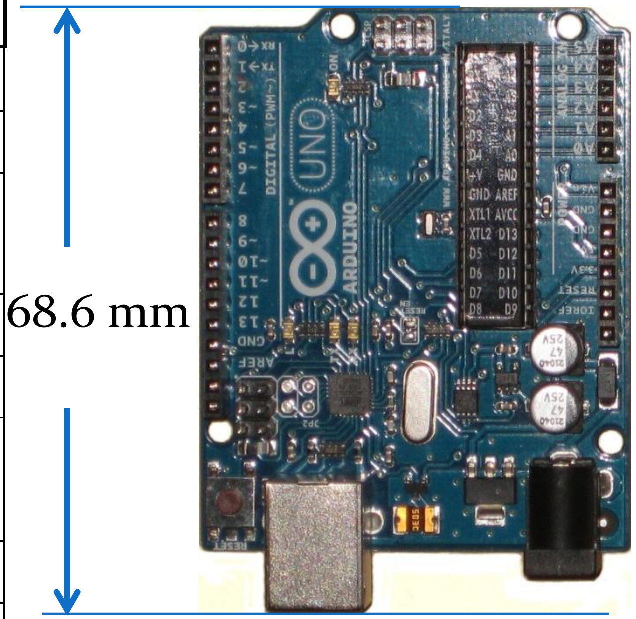


Image: Terasic Technologies

Features of MIPS 32 on DE0-Nano vs. Arduino Uno



Criteria	MIPS 32 on DE0-Nano	Arduino Uno
Clock Speed	50 MHz	16 MHz
Architecture Type	32-bit, pipelined	8-bit, single-instruction
Instruct. Memory	16-32 MB, volatile	32 kB, non-volatile flash[1]
Dynamic Memory	16-32 MB SDRAM	2 kB DRAM
EEPROM	2kB	1kB
# of digital pins	81 bidirectional + 7 input	14 digital pins, ICSP header
# of analog pins	8 analog inputs	6 analog inputs
I/O voltage	3.3V	5V
Price	$79 ($61 academic)	$25
Power draw	~180 mA (measured)	~17[2] mA, can enter sleep mode



Images: Nicole Hamilton

Programming the Processor

Toolchain for programming the processor in C

- Based on GCC, the GNU Compiler Collection
- Built a cross-compiler on Windows to make MIPS code
- Built compiler using Cygwin, a BASH terminal for Windows
- Use GCC cross-compiler to build the programs themselves

Images: GNU Foundation, Cygnus Solutions

SDRAM Controller

Original source code did not support the SDRAM on the DE0-Nano

Used Block RAM instead:

- 64 kB
- Memory is very fast (1-3 cycles)
- Implemented with registers on the FPGA

The SDRAM chip on the DE0 holds 32 MB of memory

- Previous team designed a controller for it
- Instructions stored in one 16 Mb space
- Data stored in the other 16 Mb space
- Reading/Writing takes a lot of clock cycles

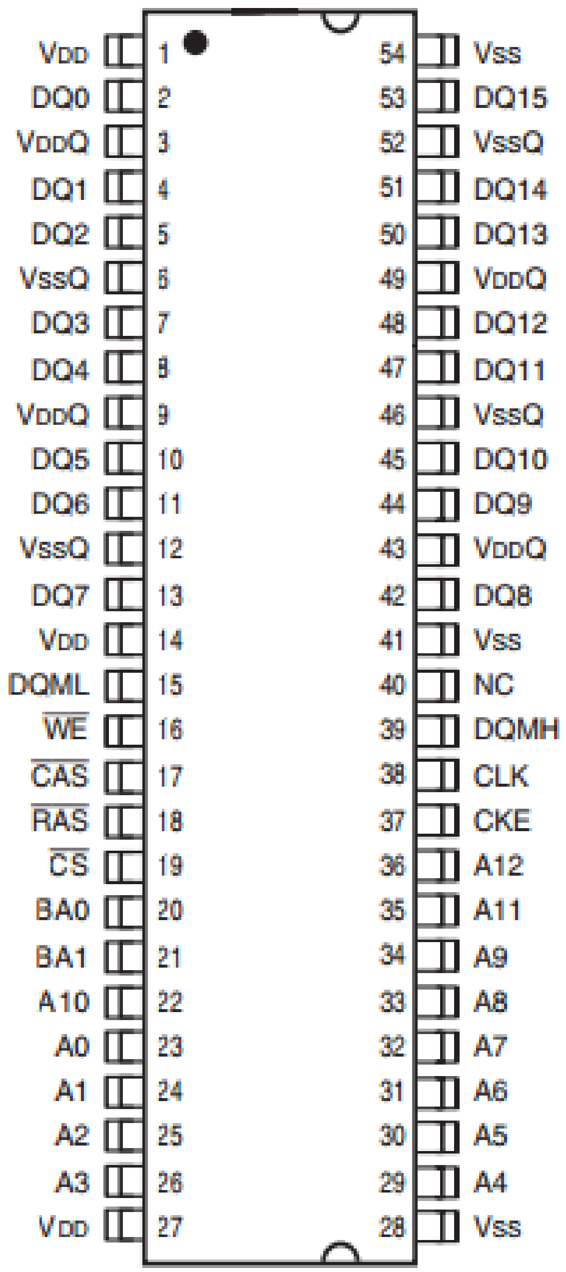


Image: Integrated Silicon Solutions, Inc.

Memory Mapping

- SDRAM provides only 32 MB, a small fraction of the 32-bit address space
- We fill in the address space with other I/O devices

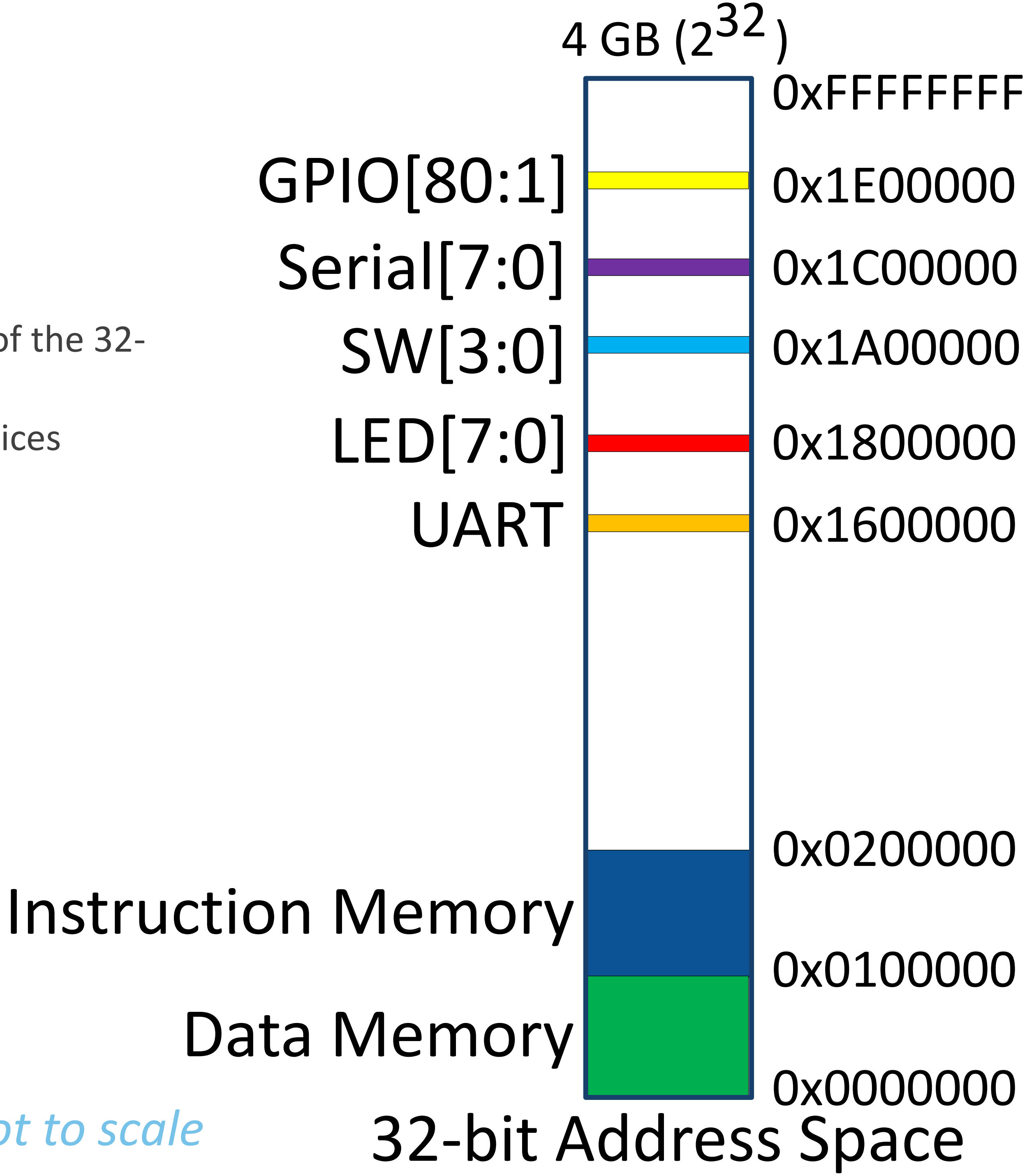
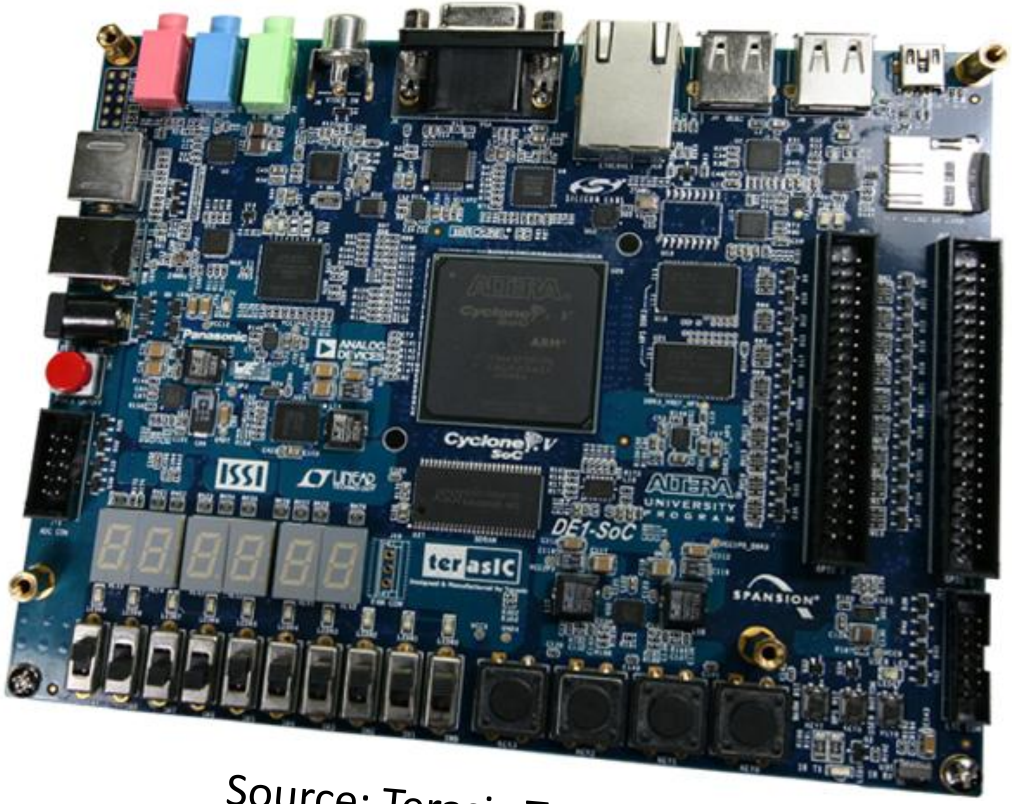


Diagram not to scale

Motivation for Top Board

The Terasic DE0 is a larger board

- Has 8 switches, 4 keys, and 6x 7-segment displays
- Ours has 8 switches, 4 keys, and 8x 7-segment displays
- Similar, but miniature

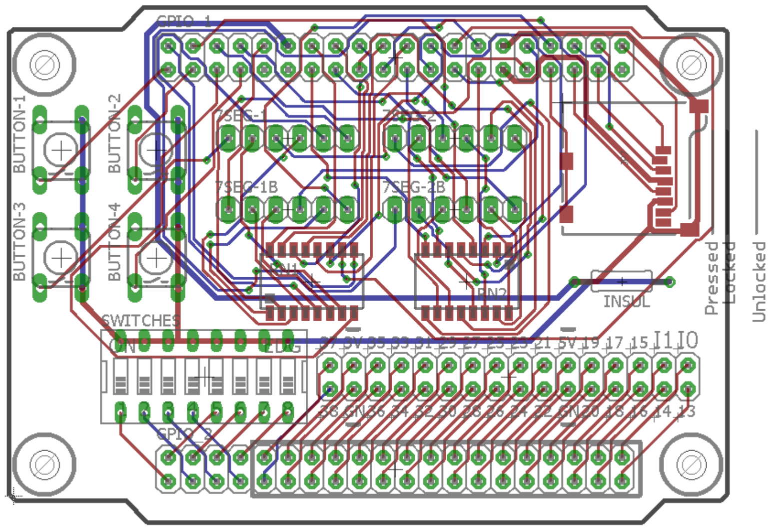


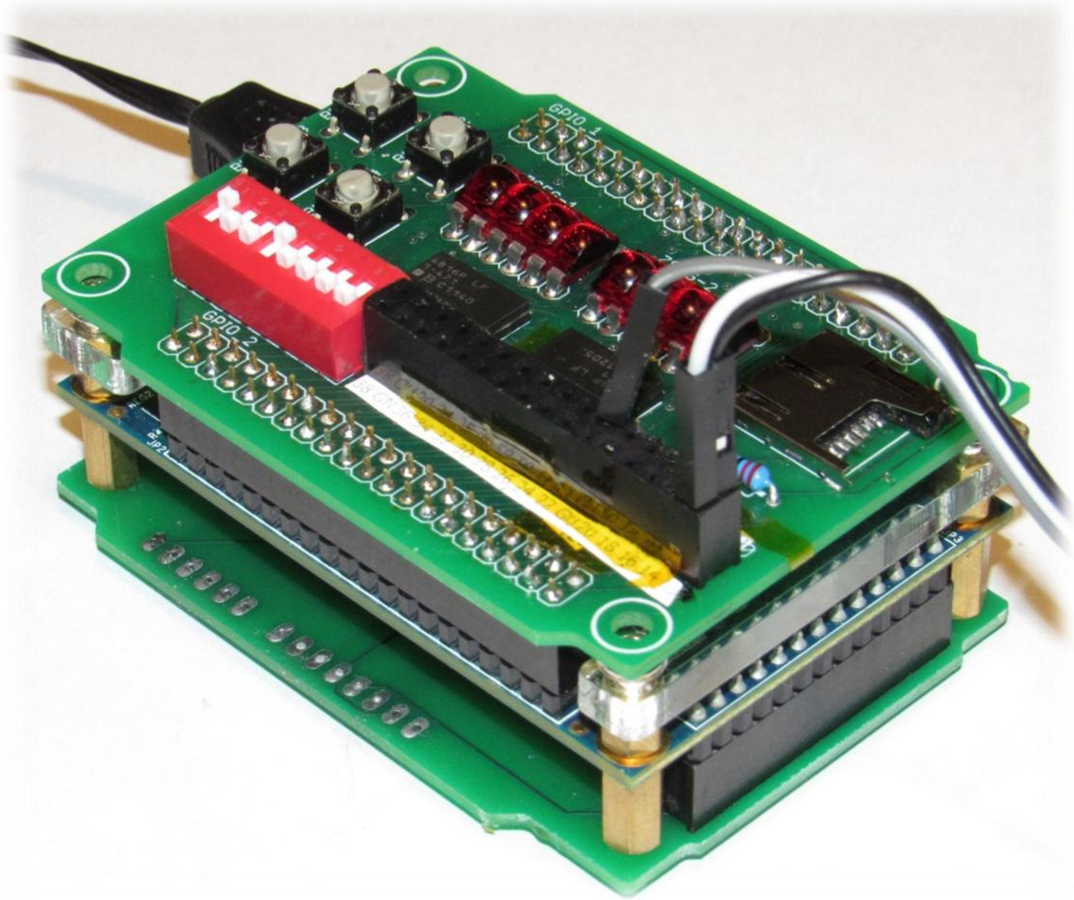
Source: Terasic Technologies

Top Board (User Interface)

I/O Devices include:

- 7-Segment displays
- Buttons
- Switches
- MicroSD Socket





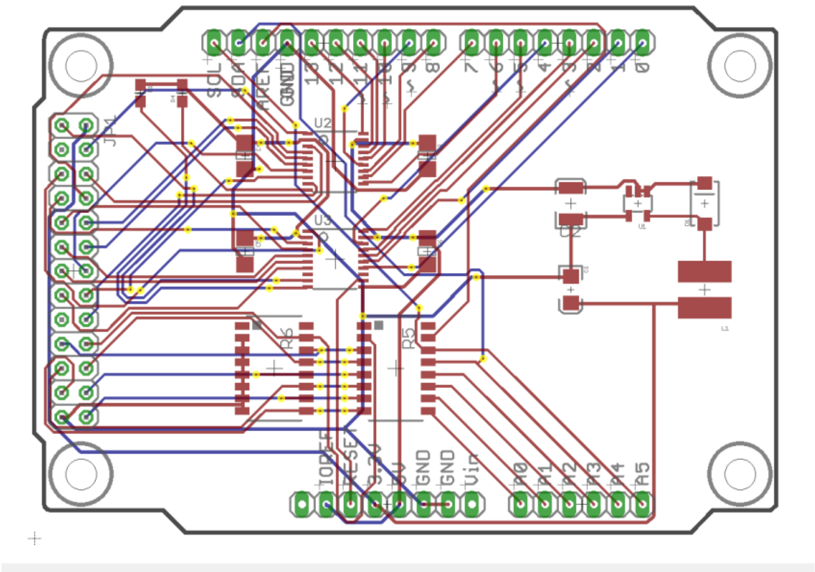
CAD program used: EAGLE

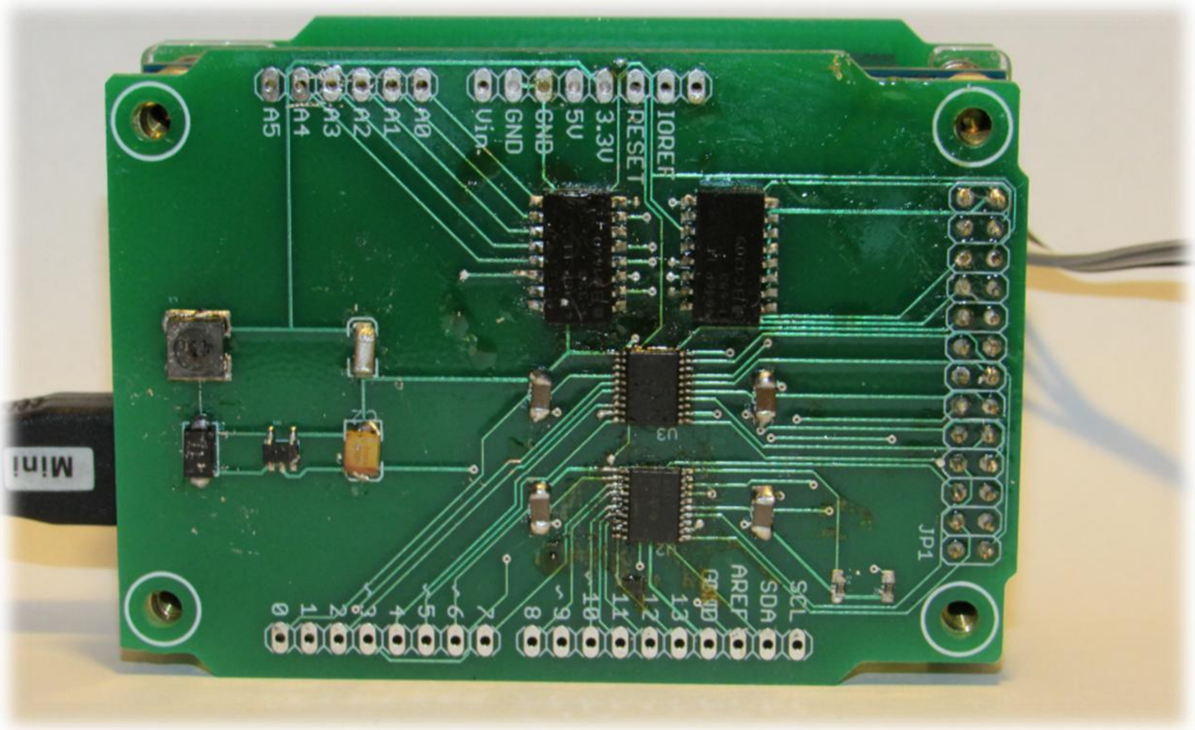
Bottom Board (Arduino Adapter)

Features:

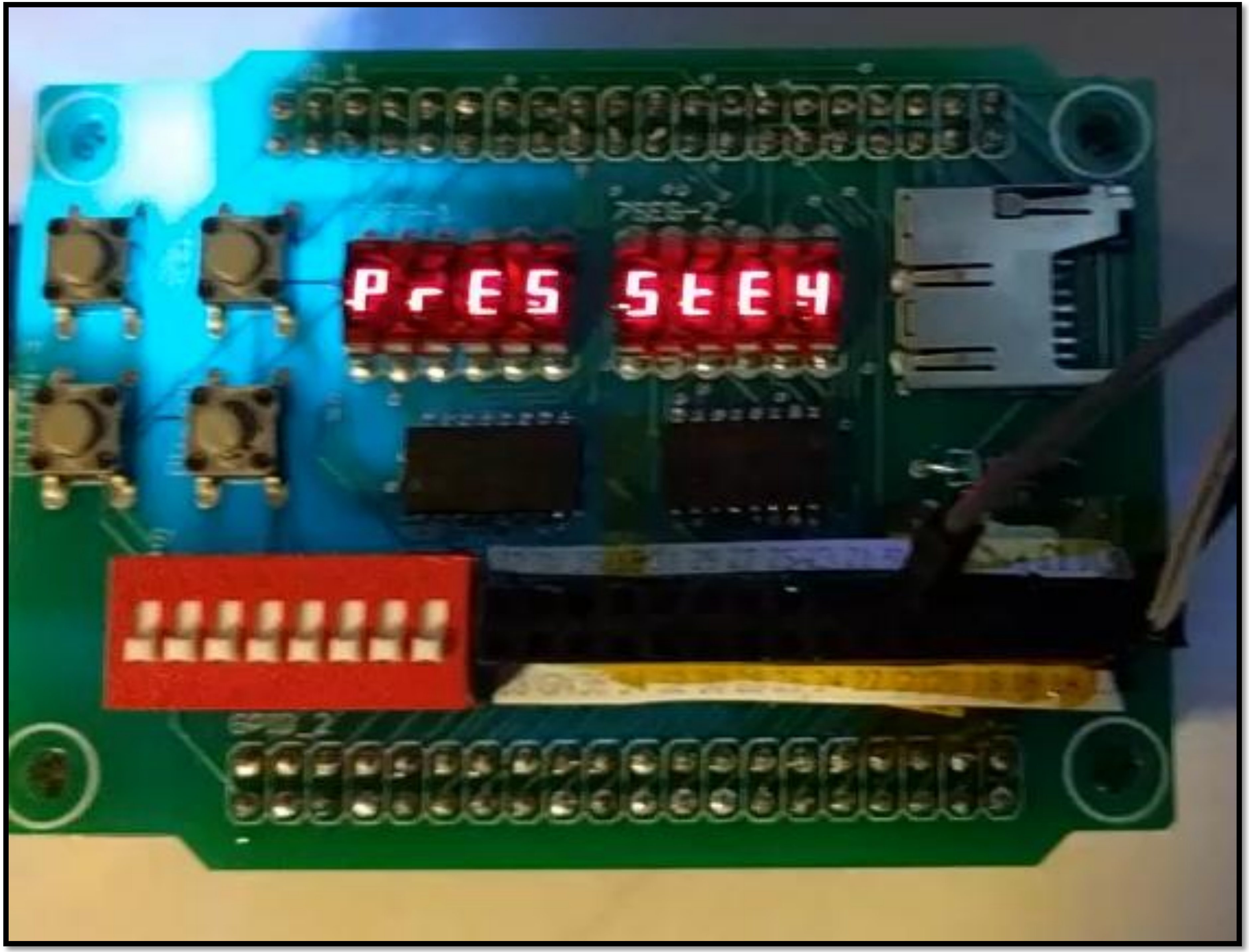
- 3.3V -> 5V Step-up
- 3.3V/5V level converters for 13 pins
- 5V -> 3.3V analog voltage dividers

  (Values 33 kΩ & 68 kΩ)



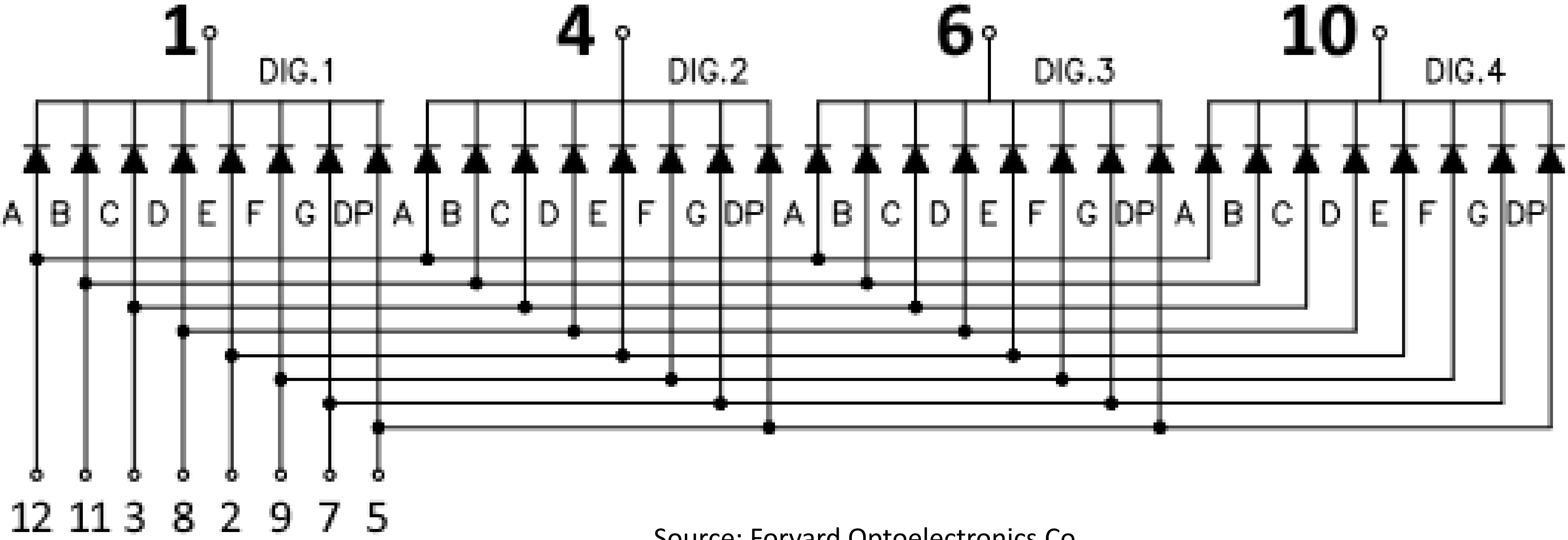


CAD program used: EAGLE


PrES StE4

7-Segment displays

The 7-segment displays are controlled by lighting up one number at a time (multiplexing)

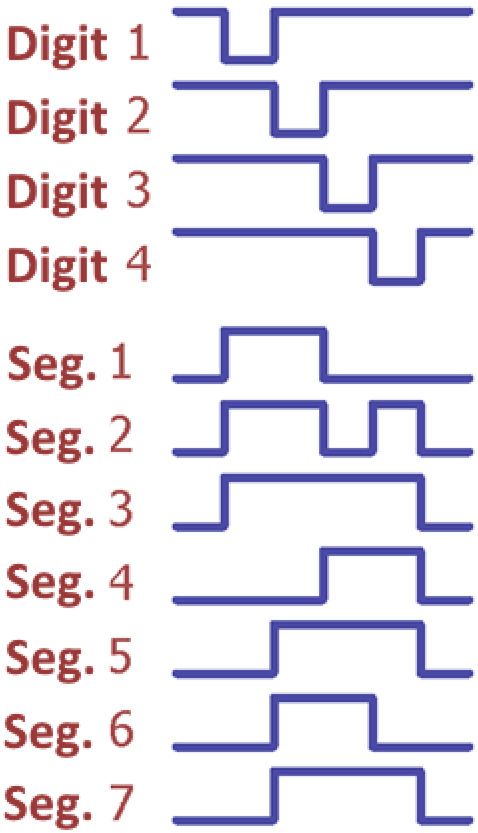


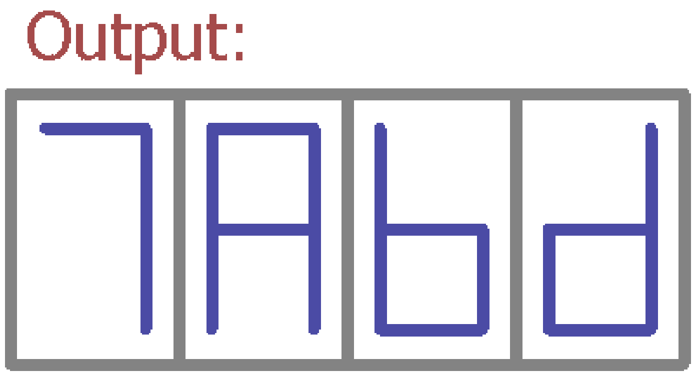
Source: Foryard Optoelectronics Co.

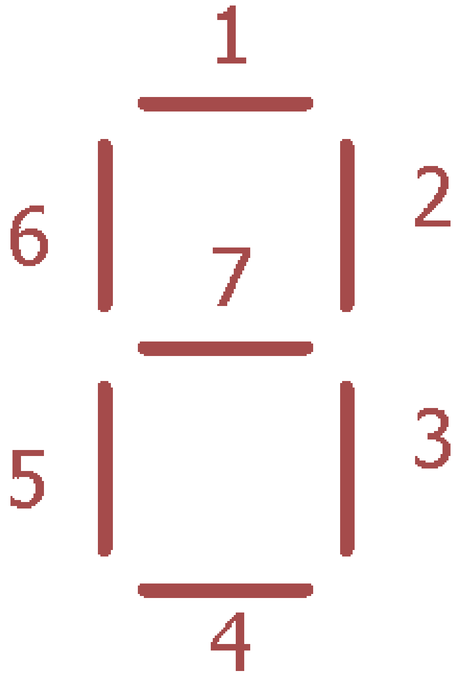
7-Segment displays

The 7-segment displays are controlled by lighting up one number at a time (multiplexing)

Example Waveform :

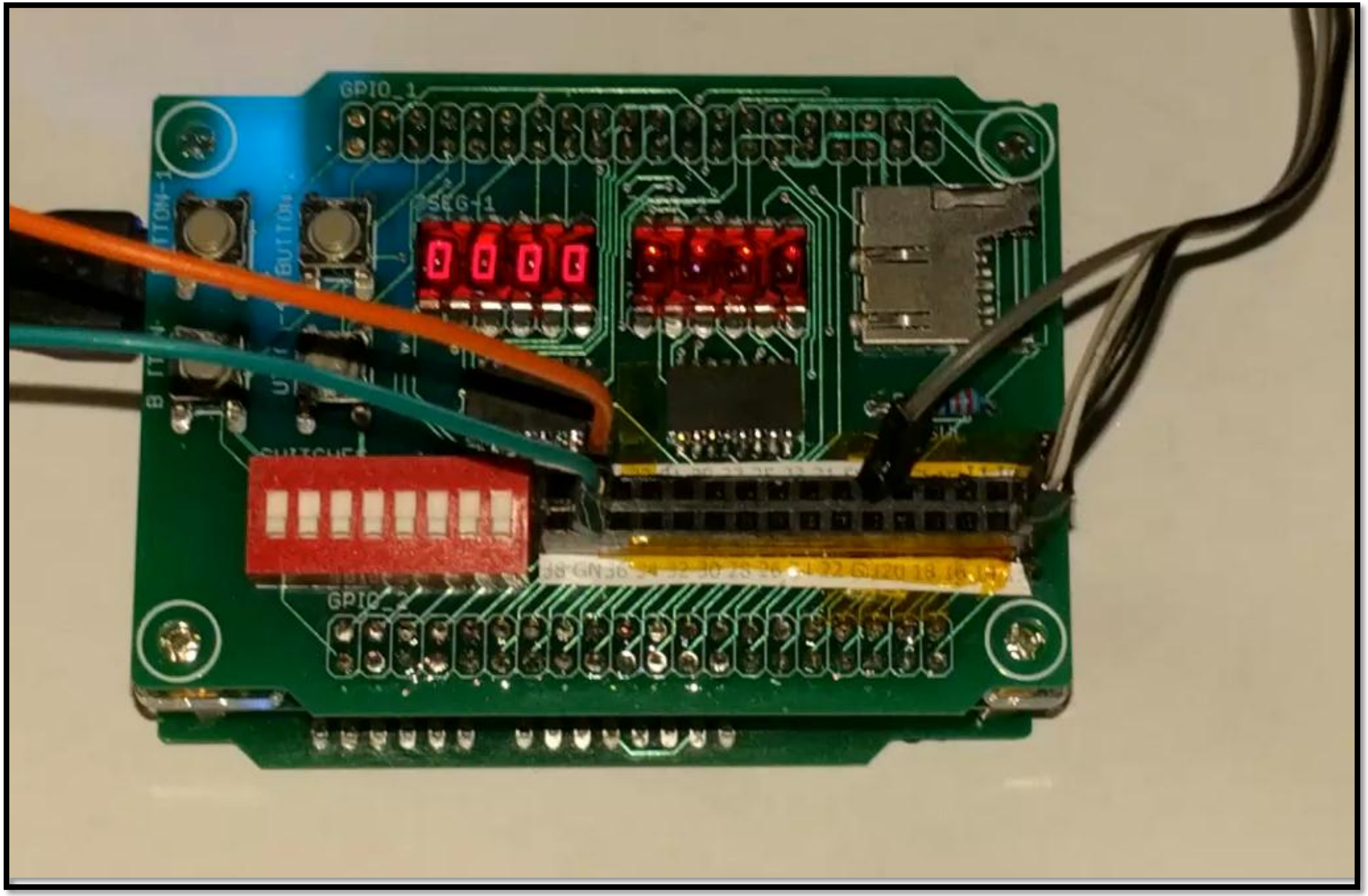






Example Code

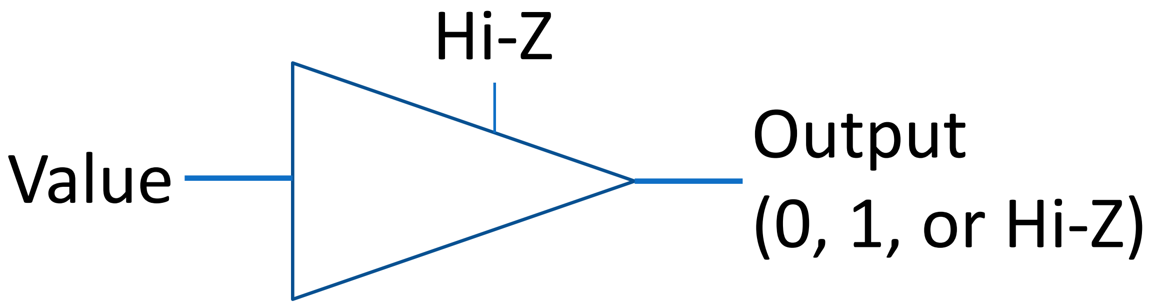
```
// Check if the first key is pressed (KEY1 or IN2)
if (!digitalRead(KEY1)) {
    if (hold == 0) {
        // Print to the 7segs
        char str[] = "FABU";
        char str2[] = "LOUS";
            setHex(0, str);
            setHex(1, str2);
    }
    hold = 1;
}
```


GPIO_1
7SEG-1
BUTTON-1
GPIO_2

I/O Input

GPIO module controls all GPIO pins, including 7-segment drivers

- GPIO pins are Tri-State, with Hi-Z, output LOW, and output HIGH
- 7-segment multiplexing module outputs a 500Hz waveform
- Hardware de-bounced inputs on buttons and switches



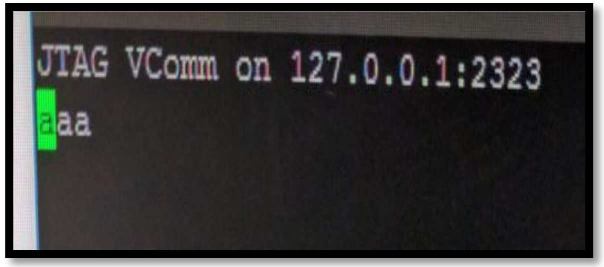
Demonstration: TeleType Console

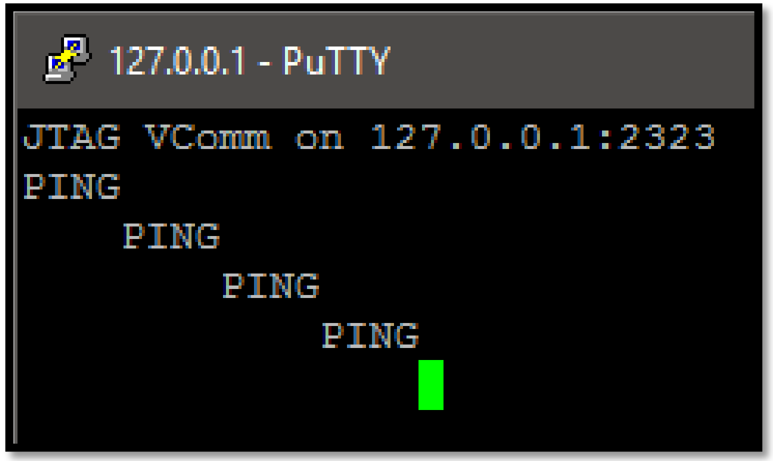
Serial module interfaces with a virtual UART module

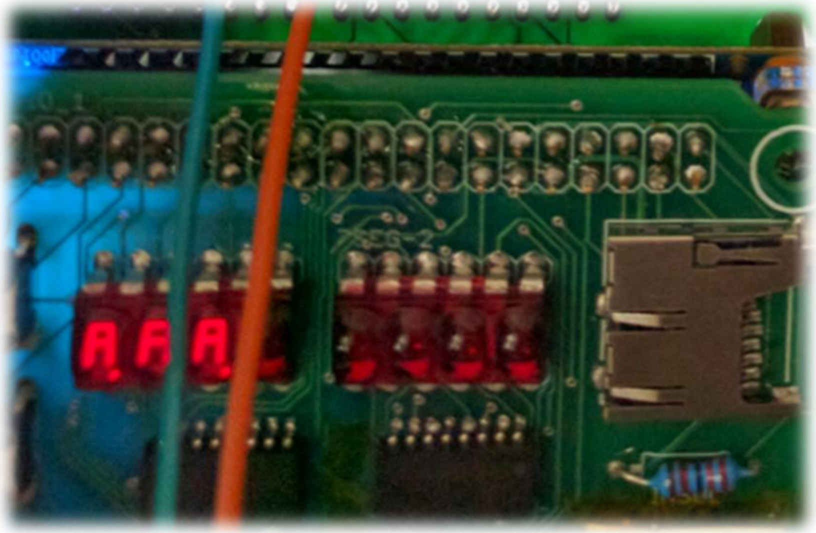
- Can send bytes to PuTTY where they will show as characters
- Can receive bytes from PuTTY (and send them back)

Example program: Console

- Press a key to write ‘PING’ to PuTTY
- Enter characters into PuTTY to display them on the 7-segment







Difficulties Encountered

Shorter Quarter

- Required a strong work ethic

No SMD soldering experience

- Learned through videos and trial/error

A lot of unknown information

- How to send bytes through the USB
- How to program in C, use pointers to access memory and I/O
- How to use Cygwin, PuTTY, and other tools
- Verilog features such as loops and variable indices

Lack of prior documentation

- Nobody has done this before and posted it online

Next Stages

~~Phase I: Create SDRAM memory controller~~ (done by last group)

~~Phase II: Integrate SDRAM controller and design I/O drivers + PCB~~ (done by us)

Potential Next Steps:

- Create an analog pin driver for the bottom board
- Create Hardware driver for MicroSD socket (using Verilog)
- Integrate with the Arduino IDE
- Implement Block-RAM memory cache for faster processing
- Add debugging functionality (IEEE Nexus standards)

Acknowledgements

Previous Group Members:

- Kevin Chao
- Derek Blankenburg
- Toan Nguyen
- Ramiro Garcia

Nicole Hamilton – Invaluable help and advising

Eric Lindberg – Sponsorship

Questions?

Thank you!

References

[1] Global Market Insights. (2016, February). Field Programmable Gate Array Market Size By Application, Industry Analysis Report, Regional Outlook, Application Potential, Competitive Market Share & Forecast, 2015 – 2022 [Online]. Available: https://www.gminsights.com/industry-analysis/field-programmable-gate-array-fpga-market-size

[2] Grand View Research. (2015, May). FPGA Market Analysis By Application And Segment Forecasts To 2020 [Online]. Available: http://www.grandviewresearch.com/industry-analysis/fpga-market

[3] MarketsandMarkets. (2016, February). FPGA Market by Architecture, Configuration, Application, and Geography - Trends & Forecasts to 2022 [Online]. Available: http://www.marketsandmarkets.com/Market-Reports/fpga-market-194123367.html

[4] ReportLinker. (2015, March). FPGA Market - Global Industry Analysis, Size, Share, Growth, Trends and Forecast, 2014 - 2020 [Online]. Available: http://www.reportlinker.com/p02878127-summary/FPGA-Market-Global-Industry-Analysis-Size-Share-Growth-Trends-and-Forecast.html