# Progress Presentation-I

e-Yantra Summer Intership-2016
Testing Embedded C Code, Spark-V Module Development, and Wireless Programming of AVR Microcontrollers

Vinay Manjunath
Mentor: Parin Chheda

IIT Bombay

June 6, 2017

# Overview of Project





Give following details:

- Project Name:
  Testing Embedded C Code, Spark-V Module Development, and Wireless Programming of AVR Microcontrollers
- Objective:
  1. Develop a Test Enviornment for Embedded C Programs
  2. Develop a Method to Wirelessly Program the SPARK V
  3. Develop a Video Tutorials for the SPARK V
- Deliverables:
  1. Automated Test Enviornment for AVR C Programs
  2. Wireless Programming Facility of SPARK V
  3. Generic wireless Programming of a 40 pin Controller
  4. Video tutorials for SPARK V

# Overview of Tasks for SPARK V Tutorials



| Tasks | Description | Deadline |
|---|---|---|
| Task 1 | Explore test suite for C viz. Unity, CMock, Ceedling | 2 days |
| Task 2 | Write different Test cases for Sample C Programs | 1 day |
| Task 3 | Unit Testing for Basic I/O Programs | 2 days |
| Task 4 | Unit Testing for PWM and Interrupt based Programs | 2 days |
| Task 5 | Develop Python Scripts to Evaluate the Programs | 1 day |
| Task 6 | Develop Test Cases for the TBT of FireBird V | 4 days |
| Task 7 | Developing Guidelines to Writing Testable Programs | 1 day |
| Task 8 | Develop a similar enviornment for Windows | 1 day |

# Overview of Tasks for SPARK V Tutorials



| Tasks | Description | Deadline |
|---|---|---|
| Task 1 | Complete Tutorial for I/O in SPARK V | 5 days |
| Task 2 | Developing Tutorials for PWM and Interrupts for SPARK V | 3 days |

# Overview of Tasks for Wirelessly Programming AVR Controllers



| Tasks | Description | Deadline |
|---|---|---|
| Task 1 | Understanding ISP and Working of Bootloaders | 2 days |
| Task 2 | Writing a simple bootloader for Atmega 16 using Zigbee | 4 days |
| Task 3 | Writing a simple bootloader for Atmega 16 using Zigbee | 4 days |
| Task 4 | Wireless Programming of SPARK V | 1 day |
| Task 5 | Generic Programming circuit for a 40 pin controller | 2 days |
| Task 6 | Documentation | 3 days |

# Tasks Accomplised



- Explore test suites for C viz. Unity, CMock, Ceedling:
  1. The three test open source projects were researched and explored.
  2. Multiple programs were used to understand how each of the test open source projects work.
- Write different Test cases for sample C Programs:
  1. Unity and its built in functions were used to test various programs and functions.
  2. It was discovered that found to test the main function of any program, that it was required to rename it.
  compiled properly.

# Tasks Accomplised



- Unit Testing for Basic I/O, ISR, PWM Programs:
  1. Editing the Avr-IO Header file to assign the registers to RAM so it can be written to and its content verified.
  2. Since the programs are not compiled using AVR-GCC the delay functions have to be mocked.
  3. To test the functionality of an interrupt its interrupt vector has to be called in a function in the main program.
  4. The method of testing of PWM based programs still needs to be explored.

# Tasks Accomplised



- Developing Python scripts for parsing and development of test cases for the 6 TBT of FireBird V:
  1. Wrote a Python script to parse the submitted document and render it testable.
  2. Integrated the script and a make file and with a final level of parsing to produce a automated test.
  3. Automated tests have been developed for Simple Motion Controll, Buzzer, and Timer Programs for the FireBird V
  4. The tests for the reminaing ADC, White line follower, and LCD programs are currently being devloped.

# Challenges Faced



- Getting Unity to work with native builds.
- Compiling the AVR C code with normal GCC.
- Generating Mock functions and intergrating them with the submitted program.
- Modification of the Main C Program to render it testable via parsing.

# Future Plans





- Guidelines for Writing Unit Tests and Automated an Test Enviornment for Windows.
- Spark V Tutorials.
- Simple Bootloader for Wireless Programming of Spark V.

# Thank You





THANK YOU !!!