

# CPRE 288 Embedded Systems Platform Movement and Code Optimization SDMay20-04

Jacob Aspinall, Geonhee Cho, Jisoo Han, Issac Klein, Sam Rai, Nathan Nordling

Client/Advisor: ISU 288 Dr. Phillip Jones, Dr. Diane Rover, Matthew Post

## Problem Statement

The current git repository for CPRE 288, tit has not been reviewed thoroughly since the labs were created. There are several known issues with the code, and potentially more that need to be fixed.

## Solution Statement

- Review of the code, and to make sure each lab is updated to reflect the current requirements
- Develop a series of automated unit tests and connect them to our Git Repository for continuous integration.
- Need to build a simulation of the external hardware(i.e. infrared sensor).

## Intended users and uses

The result will be a git repository that will be helpful for teachers, students, and will be able to handle further developments of the course.

## Functional Requirements

- Automated unit tests for all currently used code libraries/lab solutions
  - Simulated hardware(sensors, Roomba, etc.) in order to run tests automatically
  - Connected with Gitlab Continuous Integration tools
- Git repositories to be used for the CPRE 288 class
  - One for professors, TAs, ETG
  - One for students

## Non-functional Requirements

- Tests should be easy to run
- Code should be well written and easy to understand
- Git repositories should be well designed and easy to use

## Operating environment

- Microcontroller must connect with PC and continuous integration is connect with the git runner.

## Technical Details

We are creating a simulation of all the external hardware(IR sensor, sonar sensor, etc.). This allows our code to interface with “the outside world” in an automated way. We won’t need to manually retest everything after a code change.

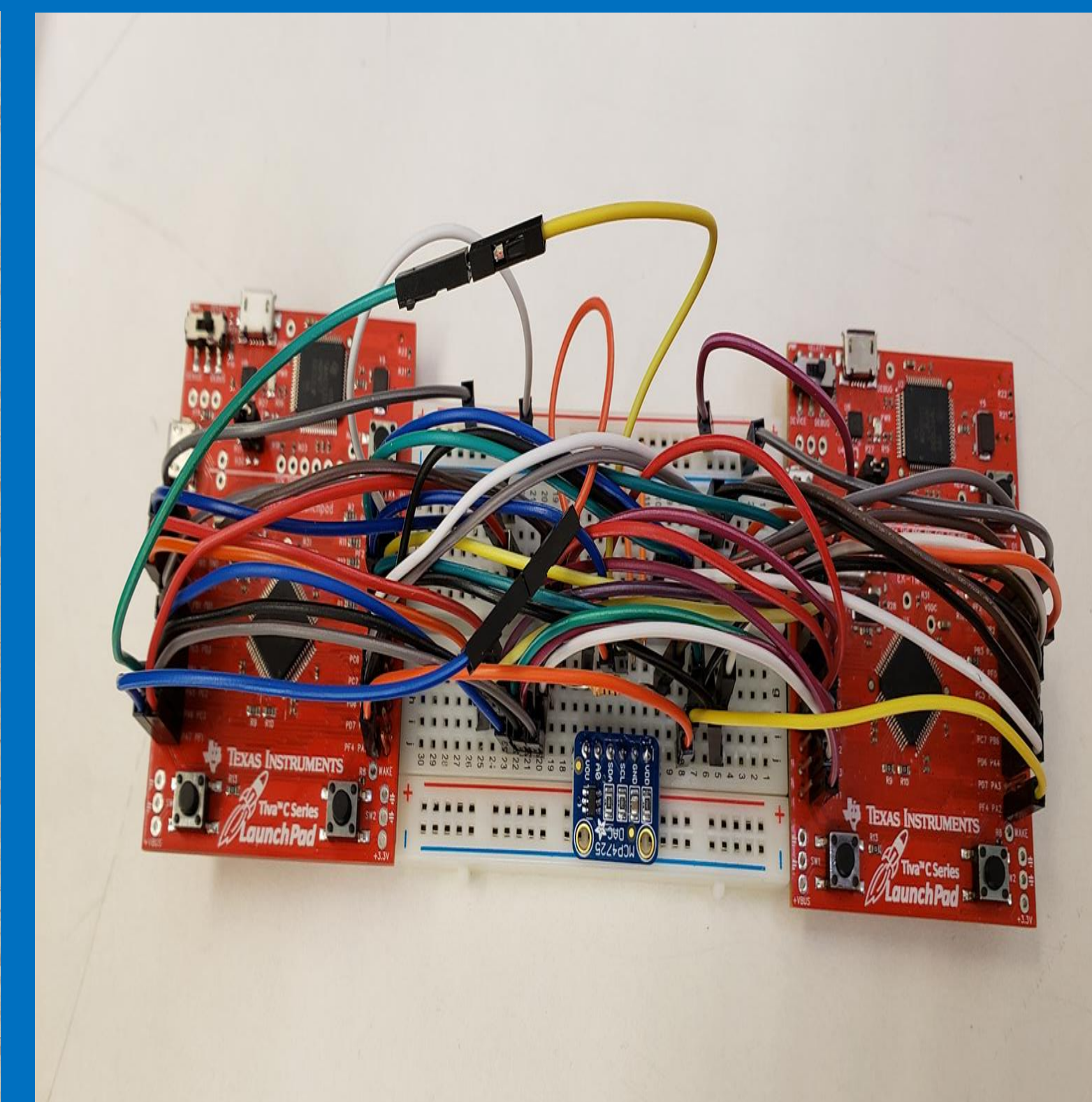
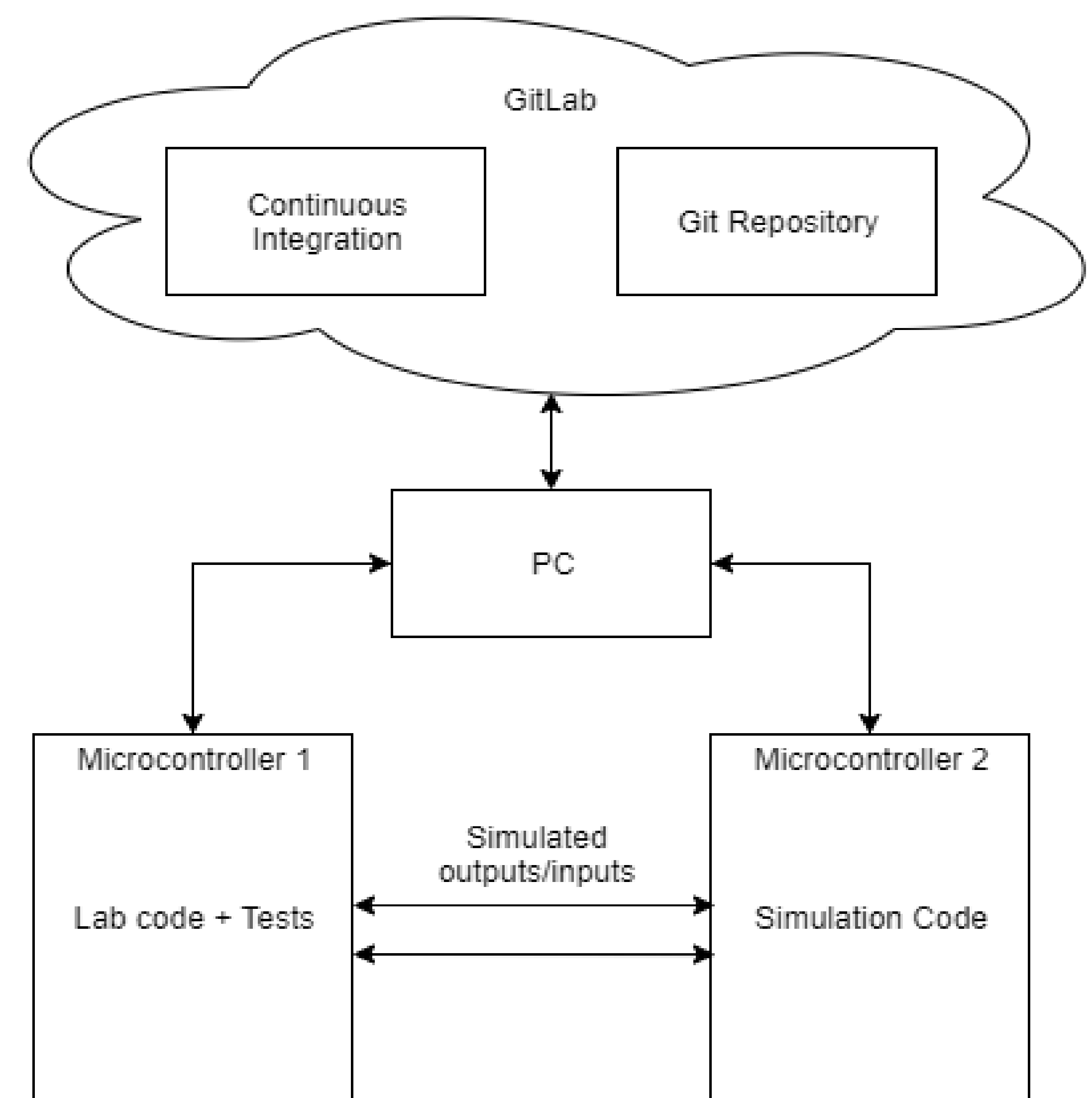
## Hardware & Software

- Code Composer Studio(CSS)
- Debug Server Scripting(DSS)
- Simulation of External Hardware(separate microcontroller to run simulations of each device)
- GitLab built in Continuous Integration tool

## Engineering Standards and Design Practices

- Agile software development & Unit testing,
- Continuous integration

## Concept sketch Block diagram



## Testing

### System Level Unit tests

- Tests implemented as C code running on the microcontroller
  - No specific testing framework
- System level tests
  - Tests to verify one use case for a device
- Unit tests
  - Testing all the individual parts of a system

### Automating the running of tests

- **Debug Server Scripting(DSS)**
  - Allows us to control our IDE’s debugger from a script
  - Lets us build our project and run it on the microcontroller automatically

### Continuous Integration

- Automatically runs unit tests on a git commit
  - Communicates with a PC connected to the simulation platform and runs DSS script
  - Test results will be reported back to Gitlab