## Scope

The automation testing platform will have the following features:

* Be able to read and send CAN messages to the ECU being tested
* Be able to write new tests and have an output of what tests have failed and what have passed
* Be able to simulate analog sensor inputs to the ECU
* Be able to control these simulated analog inputs with the software
* Be able to run all the tests without user intervention once started

## Milestones(to be reached at the end of listed week)

* 2nd week: Send and receive CAN messages with a python script(using the PCAN)
* 4th week: Testing/Pytest infrastructure basics developed
* 6th week: Core Pytest infrastructure developed
* 8th week: Communication with testing hardware
* 10th week: Testing hardware able to simulate sensor inputs on the ECU
* 12th week: Develop test cases
* 14th week: Full system test
* 16th week: Working demo of full system on a test bench

## Notes and Design Decisions

KEY  
General Note  
Selected Decision  
Decision Required  
Needs to be addressed before moving on

### Hardware

* Will be using an Arduino mega for simulating sensor inputs into the ECU
* Communication between the computer running the regression testing suite and the Arduino.   
  Options:
  + CAN command sent to the Arduino
    - Will need CAN hardware on the Arduino
    - Will need to create a CAN DBC and standard for how messages are sent and decoded
  + Command sent over serial
    - Will need to create an encoding standard for UART messages
    - Will need to figure out how to do serial communication in python

### Software

#### Pytest

Notes

Task List

#### CANBase

Notes

Task List

* Add DBC file send ability. Be able to send can messages using the signal name determined in the DBC file
* Add feature to be able to continuously send a message at a set rate. (send and update feature)