

# Demonstration of the experimental setup

## EE380 (Control Systems)

Manavaalan  
Gunasekaran

PhD student

[manvaal@iitk.ac.in](mailto:manvaal@iitk.ac.in)

Arun Kant  
Singh

PhD student

[arunkant@iitk.ac.in](mailto:arunkant@iitk.ac.in)

Ramprasad  
Potluri

Associate Professor

[potluri@iitk.ac.in](mailto:potluri@iitk.ac.in)

Department of Electrical Engineering  
Indian Institute of Technology Kanpur

August 1, 2011



Back

Forward

Close

# Contents

1	Components of the work bench	3
2	How the components are interconnected	4
3	Use MPLAB IDE: Open project file	5
4	Use MPLAB IDE: Open program in project	6
5	Use MPLAB IDE: Insert controller code	7
6	Use MPLAB IDE: Build project	8
7	Use MPLAB IDE: Select programmer	9
8	Use MPLAB IDE: Program dsPIC30F4012	10
9	How to read data	11
10	How to store data	12
11	How to plot data	13

[Back](#)[Forward](#)[Close](#)

# Components of the work bench



FG: Function Generator, PS: Power Supply, M: Motor with encoder, HB: H-bridge, PK2: PICkit 2, MCB: Motor Control Board.

The PC is also part of the work bench.

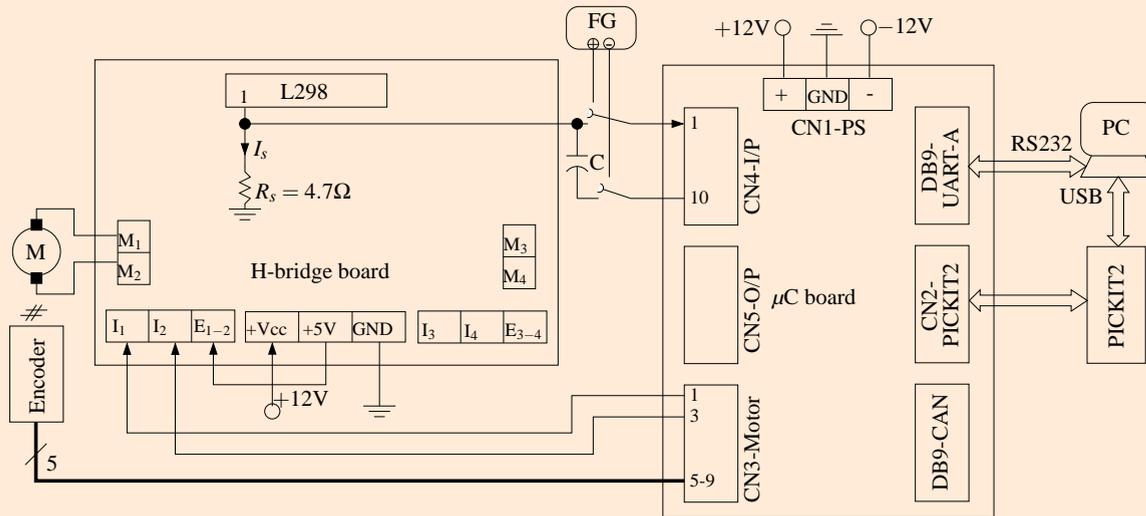


Back

Forward

Close

# How the components are interconnected



Note: The setup allows to connect another motor to H-bridge board between pins  $M_3$  and  $M_4$ . Connector CN5-O/P can then be used to connect to  $I_3$ ,  $I_4$ ,  $E_{3-4}$ . But, as we have only one QEI module on dsPIC30F4012, we cannot use encoder signal from that motor. However, we can perform speed control by using armature current sensed at pin 15 of L298 similar to how it is being sensed at pin 1.

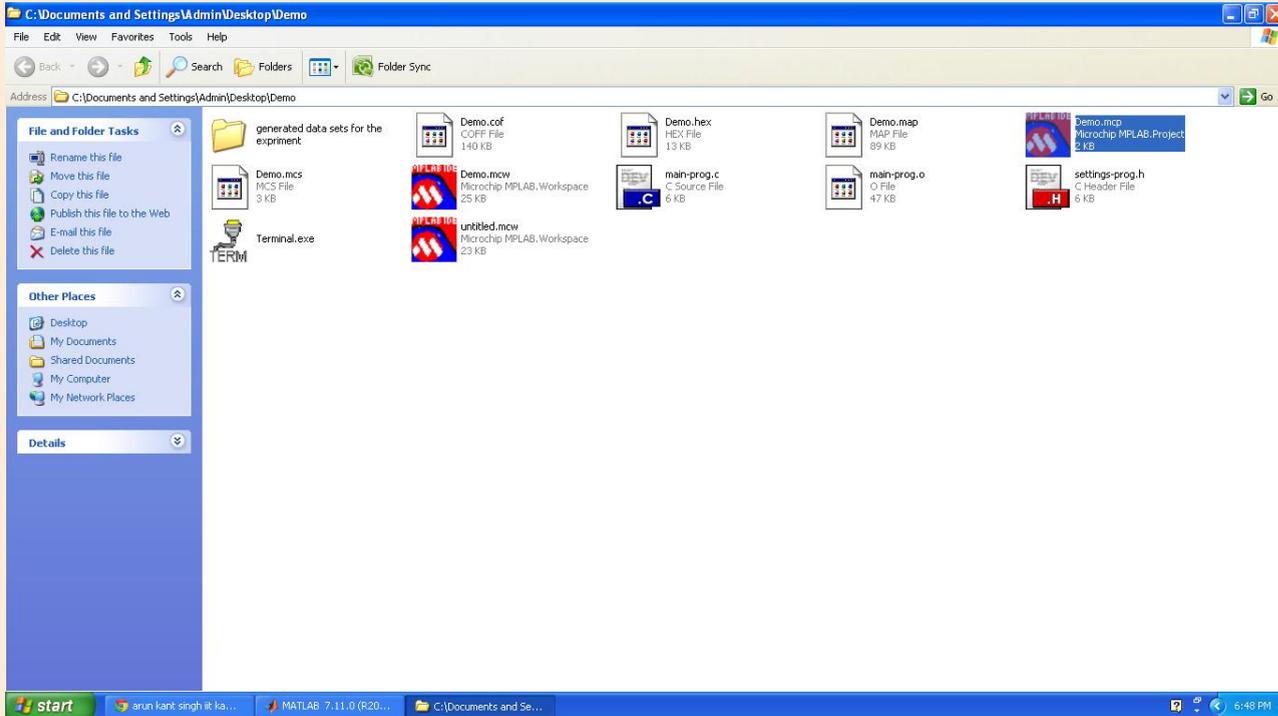


Back

Forward

Close

# Use MPLAB IDE: Open project file

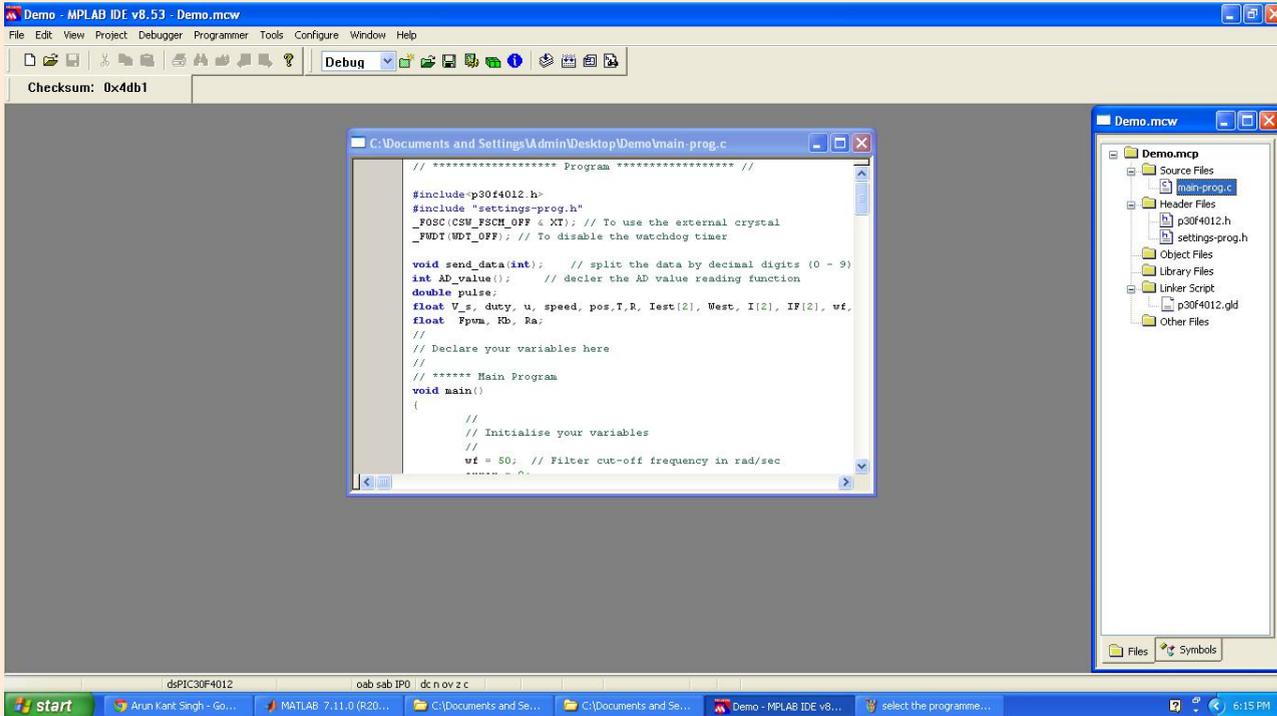


Back

Forward

Close

# Use MPLAB IDE: Open program in project



Back

Forward

Close

# Use MPLAB IDE: Insert controller code

The screenshot displays the MPLAB IDE v8.53 interface. The main window shows the source code for `main-prog.c` located at `C:\Documents and Settings\Admin\Desktop\Demo\main-prog.c`. The code implements a current sensing and PWM control loop. A specific section of the code is highlighted with a box, indicating where the user's controller code should be inserted.

```

/**
 * I[1] = AD_value(); // read the current
 * //send_data(I[1]);
 * //uart_tx(9);
 * I[1] = 5*(512 + I[1])/1024/4.7/3;
 * //I[1] = 5*(512 + I[1])/1024/4.7;
 * //Isense = 5*(512 + I[1])/1024/4.7/3;
 * IF[1] = (wf*T/(2+wf*T))*(I[1] + I[0]) + ((2 - wf*T)/(2 + wf*T))*IF[0];
 * IF[0] = IF[1];
 * I[0] = I[1];
 * // send_data(IF[1]*1000); // to send the filtered current
 * // IF[1] is the current instant of the filtered sensed current
 *
 * //***** Your controller code *****
 * //***** goes in place of this box *****
 * //*****
 *
 * //u=7; // for step input enable this to provide step of 7
 * if(u > 0.8 * V_s)
 *     u = 0.8 * V_s; // positive saturation
 * else if(u < -0.8 * V_s)
 *     u = -0.8 * V_s; // negative saturation
 * duty = u/V_s;
 *
 * pwm_con(duty); // update the PWM with respect to the new duty ratio
 * uart_tx(9);
 * send_data(u*100); // sending 100 times the controller effect u
 *
 * // ***** control ends *****
 *
 * LATBbits.LATE0=0;

```

The Output window on the left shows the following text:

```

Build Version Control Find in Files PICKit 2
Initializing PICKit 2 version 0.0.3.63
Found PICKit 2 - Operating System Version 2.32.0
Target power not detected - Powering from PICKit 2 (5.00V)
dsPIC30F4012 found (Rev 0x1004)
PICKit 2 Ready

```

The right-hand pane shows the project structure for `Demo.mcp`, including Source Files, Header Files, Object Files, Library Files, and Other Files.



Back

Forward

Close

# Use MPLAB IDE: Build project

**Output**

```
Build Version Control Find in Files
.ivt          0x4          0x
.iivt         0x84         0x
.text         0x100        0x7
.dinit        0x8b2        0:
.isr          0x8ba        0:
_FOSC         0xf80000     0:
_FWDT         0xf80002     0:

Total program memory used (byte)

Data Memory Usage
section      address      alignment ge
-----
.nbss        0x800

Total data memory used (byte)

Dynamic Memory Usage
region      address
-----
heap        0
stack      0x850

Maximum dynamic memory (byte)

Executing: "C:\Program Files\Microchip\MPLAB C30\bin\pic30-bin2hex.
Loaded C:\Documents and Settings\Admin\Desktop\Demo\Demo.ccf.

Debug build of project 'C:\Documents and Settings\Admin\Desktop\De
Preprocessor symbol ' _DEBUG ' is defined.
Sat Jul 30 18:16:51 2011

BUILD SUCCEEDED
```

**C:\Documents and Settings\Admin\Desktop\Demo\main-prog.c**

```
// ***** Program ***** //
#include "p30f4012.h"
#include "settings-prog.h"
_FOSC(CSW_FSCM_OFF & XT); // To use the external crystal
_FWDT(WDT_OFF); // To disable the watchdog timer

void send_data(int); // split the data by decimal digits (0 - 9)
int AD_value(); // decler the AD value reading function
double pulse;

float V_s, duty, u, speed, pos,T,R, Iest(2), West, I(2), IF(2), wf,
float Fpm, Kb, Ra;

// Declare your variables here
//
// ***** Main Program
void main()
{
//
// Initialise your variables
//
wf = 50; // Filter cut-off frequency in rad/sec
}

dsPIC30F4012 oab sab IPO | dc n ov z c Ln 1, Col 1 INS WR
```

**Demo.mcp**

- Source Files
  - main-prog.c
- Header Files
  - p30f4012.h
  - settings-prog.h
- Object Files
- Library Files
- Linker Script
  - p30f4012.gld
- Other Files

Files Symbols

start Arun Kant Singh - Go... MATLAB 7.11.0 (R20... C:\Documents and Se... C:\Documents and Se... Demo - MPLAB IDE v8... oping the project fil... 8:16 PM

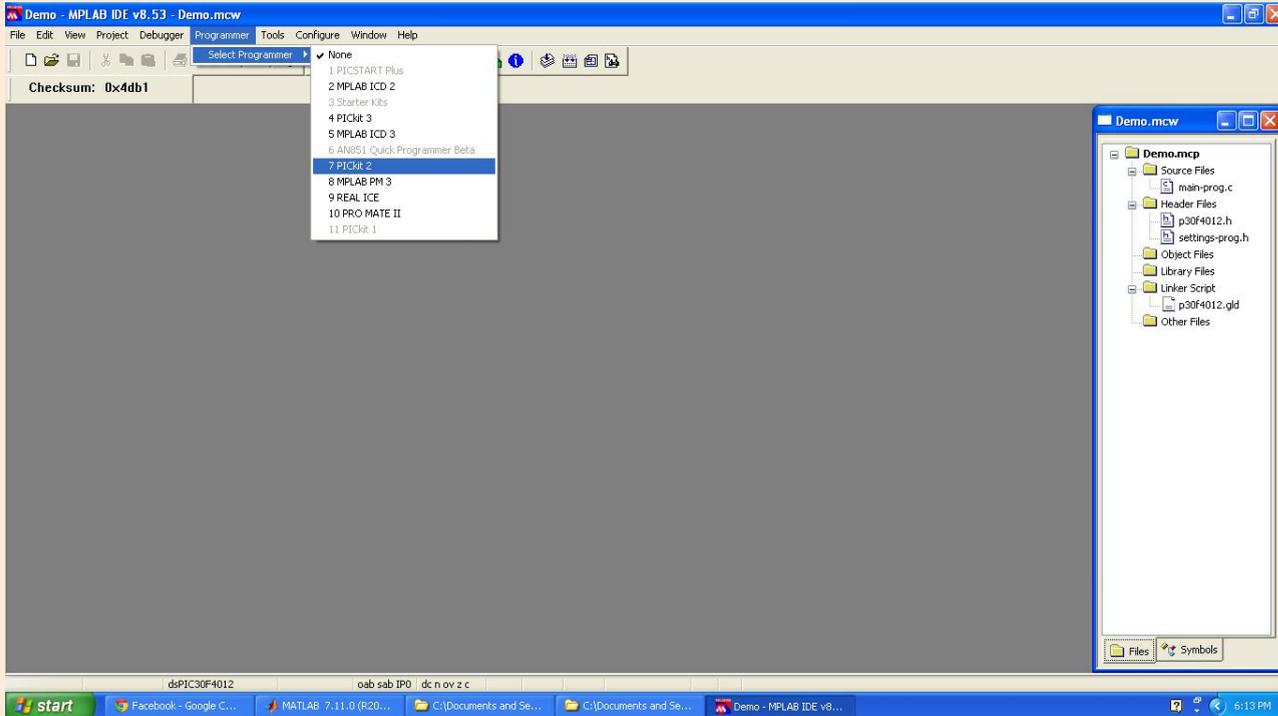


Back

Forward

Close

# Use MPLAB IDE: Select programmer



Back

Forward

Close

# Use MPLAB IDE: Program dsPIC30F4012

The screenshot displays the MPLAB IDE v8.53 interface. The top menu bar includes File, Edit, View, Project, Debugger, Programmer, Tools, Configure, Window, and Help. The toolbar contains various icons for file operations and debugging. The status bar at the bottom shows the current project is dsPIC30F4012, with tabs for PICKit 2, dsPIC30F4012, oab sab IPO, dc n ov z c, and build the project.JPG.

**Output Window:**

```
Build | Version Control | Find in Files | PICKit 2
Initializing PICKit 2 version 0.0.3.63
Found PICKit 2 - Operating System Version 2.32.0
Target power not detected - Powering from PICKit 2 (5.00V)
dsPIC30F4012 found (Rev 0x1004)
PICKit 2 Ready
```

**Source Code Window:**

```
// ***** Program ***** //
#include <pic30f4012.h>
#include "settings-prog.h"
_FOSC(CSW_FSCM_OFF & XT); // To use the external crystal
_FWDT(WDT_OFF); // To disable the watchdog timer

void send_data(int); // split the data by decimal digits (0 - 9) in 3 digit form
int AD_value(); // decler the AD value reading function
double pulse;
float V_s, duty, u, speed, pos, T, R, Iest(2), West, I(2), IF(2), wf, error;
float Fpwm, Kb, Ra;
//
// Declare your variables here
//
// ***** Main Program
void main()
{
    //
    // Initialize your variables
    //
    wf = 50; // Filter cut-off frequency in rad/sec
    error = 0;
    I(0) = 0; // Initial past current
    IF(0) = 0; // Initial past filtered current
    duty = 0.0;
    u = 0; // initialise the controller o/p
    speed = 0.0; // Rad/sec
    pos = 0.0; // radians
    T = 0.002; // Sampling time in sec
    Fpwm = 50; // Pwm Frequency in kHz
    V_s = 12.0; // Power supply Voltage
    TRISD = 0; // D port is configured as output port
    LATD = 1; // used for direction control
    qei_set(); // Initialize the QEI settings
}
```

**Project Explorer Window:**

```
Demo.mcp
├── Source Files
│   ├── main-prog.c
│   ├── p30f4012.h
│   └── settings-prog.h
├── Header Files
│   └── settings-prog.h
├── Object Files
├── Library Files
├── Linker Script
├── p30f4012.gld
└── Other Files
```



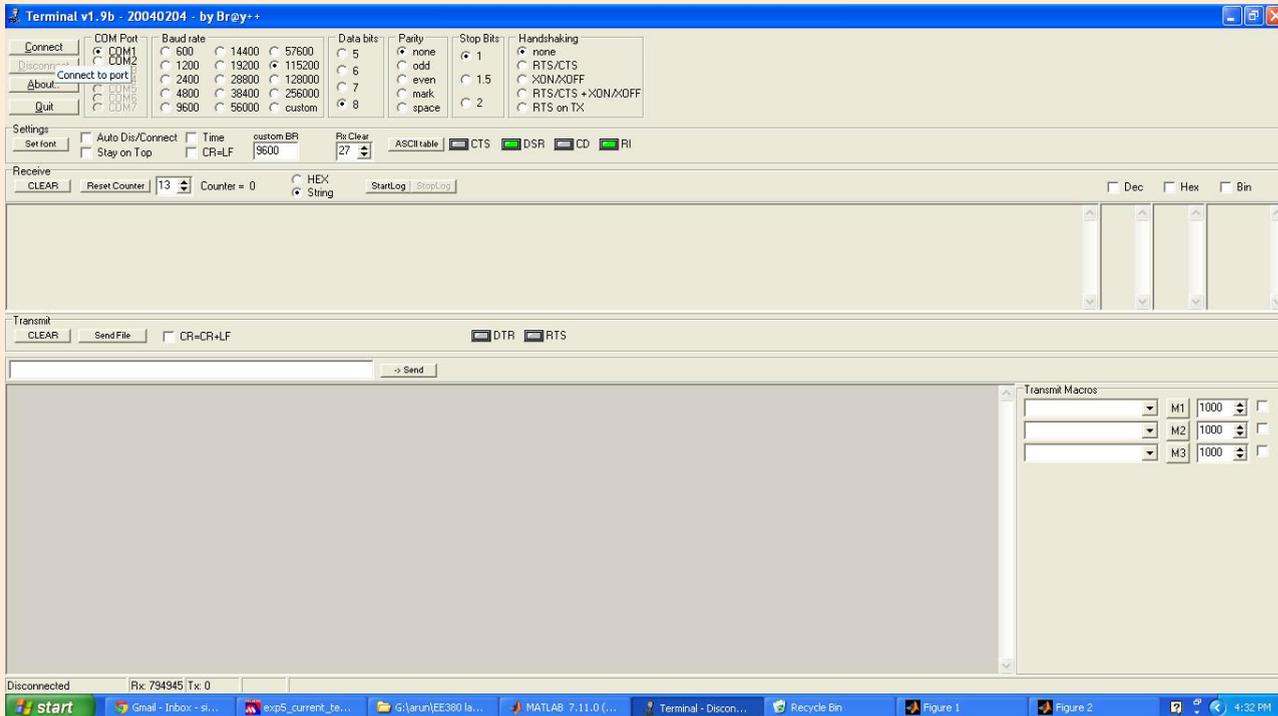
Back

Forward

Close

# How to read data

Start terminal.exe. Configure the GUI as shown.



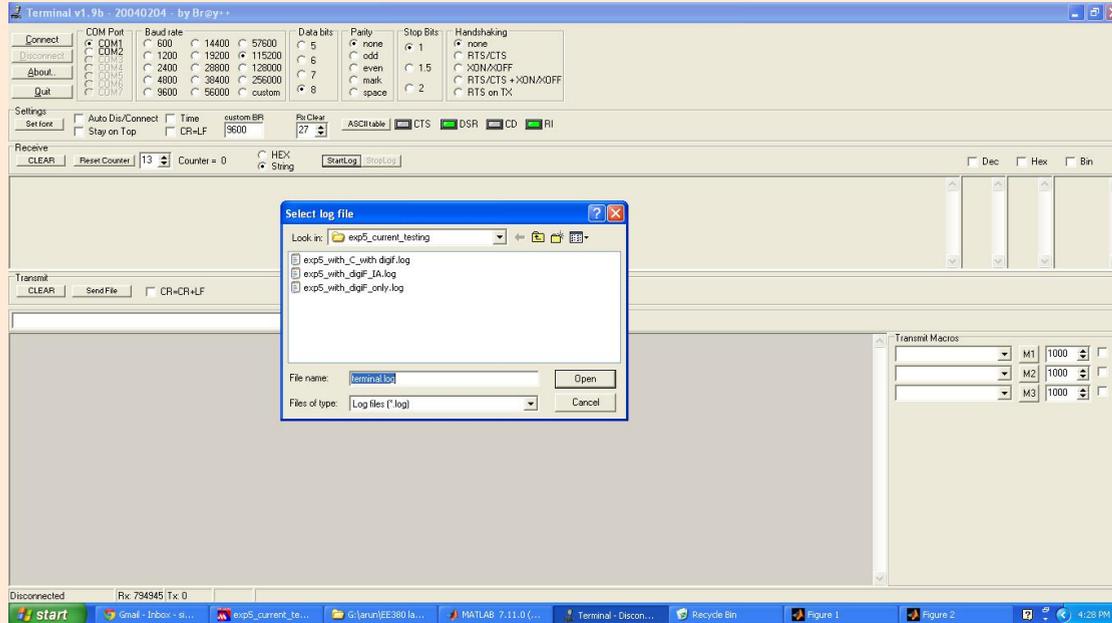
Back

Forward

Close

# How to store data

- Click on Start Log button; provide a file name and location to store this file.



- Hold down reset button on MCB; click on Connect button on terminal.exe.
- Release reset button; read in data for a little longer than  $t_s$  of your control system.
- When done, click on Stop Log button on terminal.exe.



# How to plot data

- Let log file created by `terminal.exe` be `testdata.txt`.
- Remove the parts of the text from the beginning and end of the file that is not the data about variables of interest from the  $\mu\text{C}$ .
- Place m-file `readplot.m` in folder that contains `testdata.txt`.
- Start up GNU Octave in this folder.
- Execute `readplot.m` in GNU Octave.
- If plots have problems appearing, apply correction given in `readplot.m`.



Back

Forward

Close