1. CPE 325: Getting Started with the UAH Serial Application

Objectives: This tutorial describes how to configure the UAH serial application to plot signals sent from the TI Experimenter board via a serial link.

The TI Experimenter Board runs a program (ramp_wave) that sends a 2-byte message every 32 ms. The first byte of the message is a header byte (0x55) and the second byte contains a value of an incrementing counter.

1.1. Ramp_wave Program

Figure 1 shows the ramp_wave program.

```
1.
2.  /*****************************************************************************
3.  * Description: This program sends 8-bit counter value via a UART channel.
4.  *              It illustrates the use and configuration
5.  *              of the UAH Serial Application.
6.  *              Watchdog timer is configured in the interval mode.
7.  *              It generates an interrupt request every 32 ms.
8.  *              The service routine sends a 2-byte message over UART:
9.  *              8-bit header (0x55) followed by a 8-bit counter value.
10. *              The counter is incremented every time we enter the service routine.
11. *              The UAH Serial App should plot a ramp-like signal.
12. *
13. * Platform:    TI Experimenter's board with MSP430FG4618/F2013
14. *
15. * Files:       ramp_wave.c
16. *
17. * Setup:       Connect RS-232 cable from TI Experimenter's board with
18. *              MSP430FG4618/F2013 to a workstation that plots the.
19. *
20. *
21. * Author:      Aleksandar Milenkovic
22. * Contact:     milenkovic@computer.org
23. * Date:        June 2013
24. *****************************************************************************
    */
25.
26. #include <msp430xG46x.h>
27.
28. // UART Initializaion
29. void UART_Initialize(void);
30.
31. //send char function
32. void UART_putchar(char c);
33.
34. void main(void)
35. {
36.     //WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer
37.     WDTCTL = WDT_MDLY_32; // watchdog timer in interval mode, 32 ms
38.     //Initialize the UART
39.     UART_Initialize();
40.     IE1 |= WDTIE;
```

```
41.     _BIS_SR(LPM0_bits + GIE);
42. }
43.
44. /* 38,400 bits/sec, 1 stop bit, no parity */
45. void UART_Initialize(void)
46. {
47.       P2SEL |= BIT4+BIT5; // Set UC0TXD and UC0RXD to transmit and receive data
48.       UCA0CTL1 |= BIT0;   // software reset
49.       UCA0CTL0 = 0;       // USCI_A0 control register
50.       UCA0CTL1 |= UCSSEL_2; // clock source SMCLK
51.       UCA0BR0=27;       // 1 MHz 38400
52.       UCA0BR1=0;        // 1 MHz 38400
53.       UCA0MCTL=0x94;    // Modulation
54.       UCA0CTL1 &= ~BIT0; // software resetIE2 |=UCA0RXIE;
55. }
56.
57. void UART_putchar(char c)
58. {
59.       // wait for other character to transmit
60.       while (!(IFG2 & UCA0TXIFG));
61.       UCA0TXBUF = c;
62. }
63.
64. // Watchdog Timer, interval mode, interrupt service routine
65. #pragma vector=WDT_VECTOR
66. __interrupt void watchdog_timer(void)
67. {
68.   static char cnt = 0;
69.   cnt++;
70.   UART_putchar(0x55);
71.   UART_putchar(cnt);
72. }
```

Figure 1. Ramp_wave.c.

1.2. Configuration of UAH Serial App to Display Ramp Wave

You should locate the serial COM port used to connect to the TI Experimenter Board. Open Control Panel -> Device Manager -> Ports (COM&LPT) window. Figure 2 shows that COM5 is used to connect to the boards serial port.

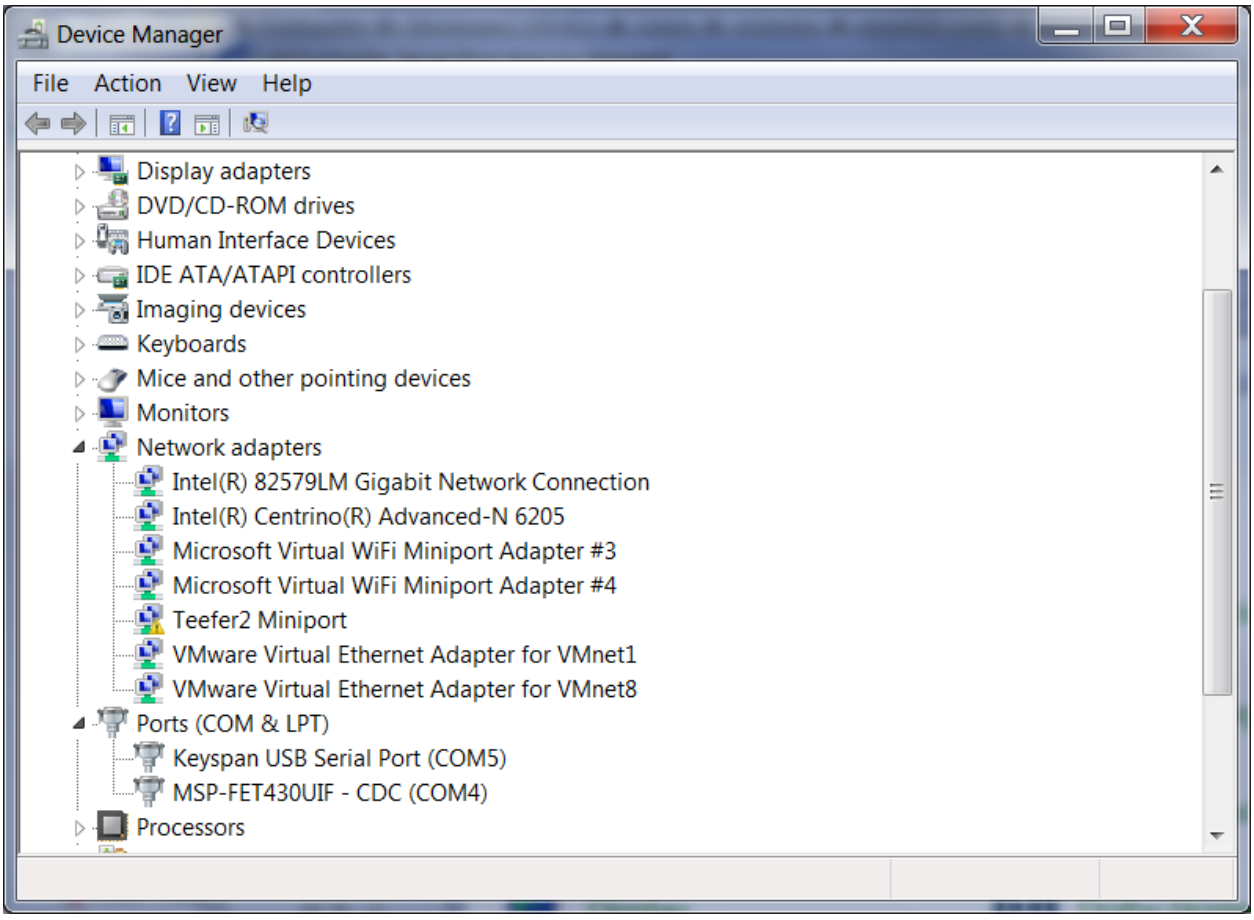

Figure 2. Locating COM ports.

Run the UAH Serial App and open up the settings tab as shown in Figure 3. Configure the parameters as follows.

- X-label: Sample Number (as we are not transmitting timestamps).
- Number of Samples to draw at the time: 1 (draw every new sample one by one)
- Number of Samples on graph = 2048
- Packet Size = 2 bytes
- Number of Channels = 1
- **CH 0 Configuration:**
    - Name: ramp
    - Type: Byte 8bit (the payload to be plotted)
    - Position: 1 (header is byte 0, payload is byte 1)
    - Show on Graph: checked
    - Y axis: Y1 checked

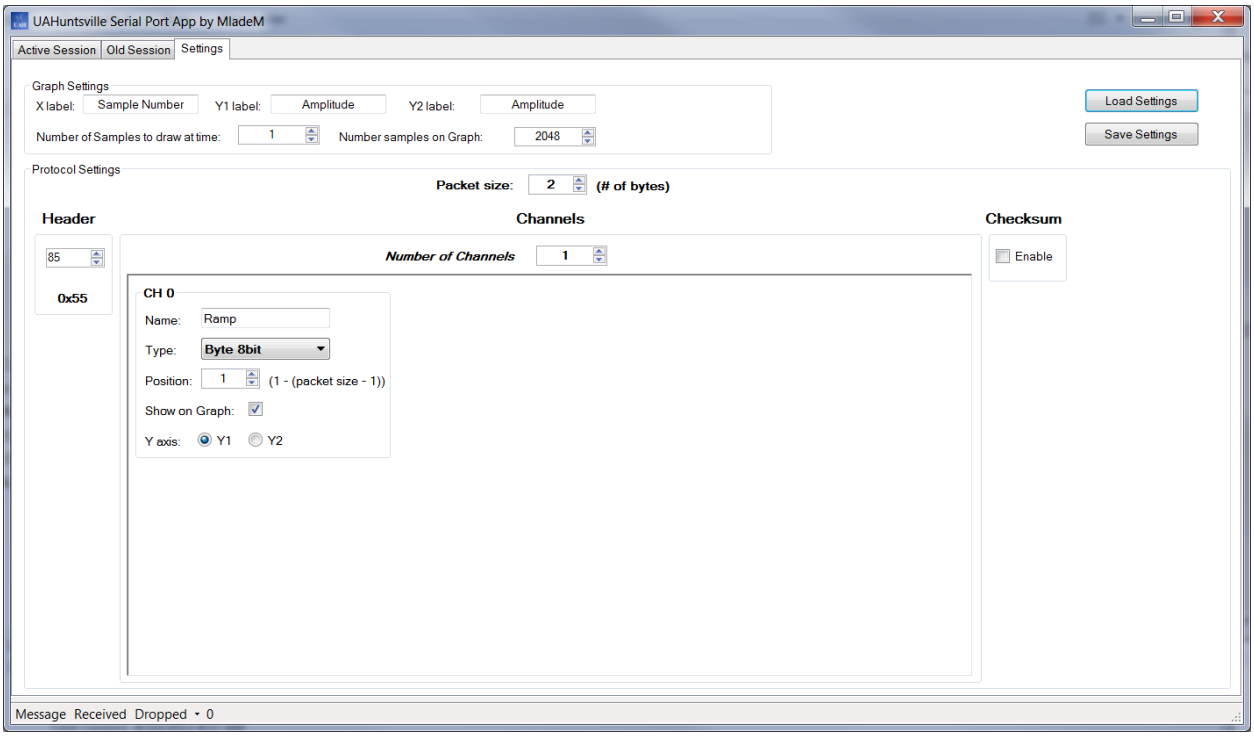

Figure 3. Settings screen configuration.

Configure active session as shown in Figure 4.

- Com port: COM5
- Baud rate: 38,400
- Data bits: 8
- Parity: no
- Stop bits: 1

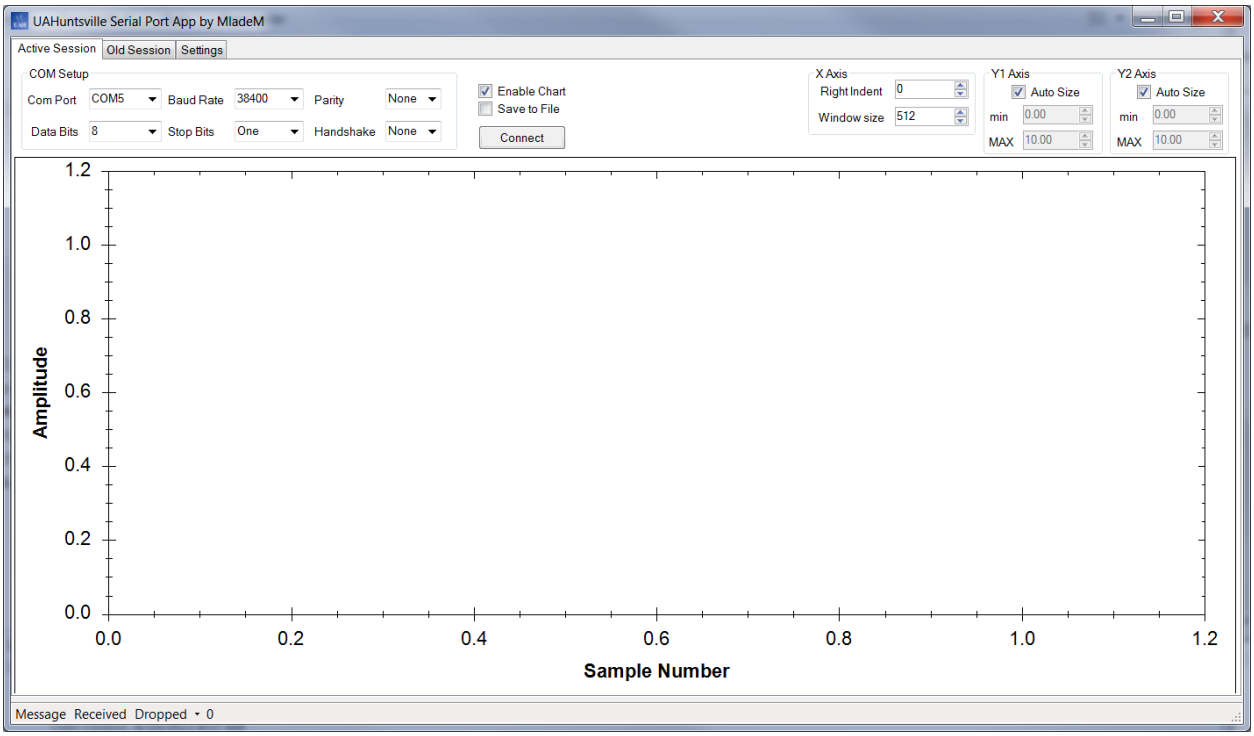

Figure 4. Session screen configuration.

Download and run your ramp_wave program on the development board. Press Connect. You should see a plot as shown in Figure 5.

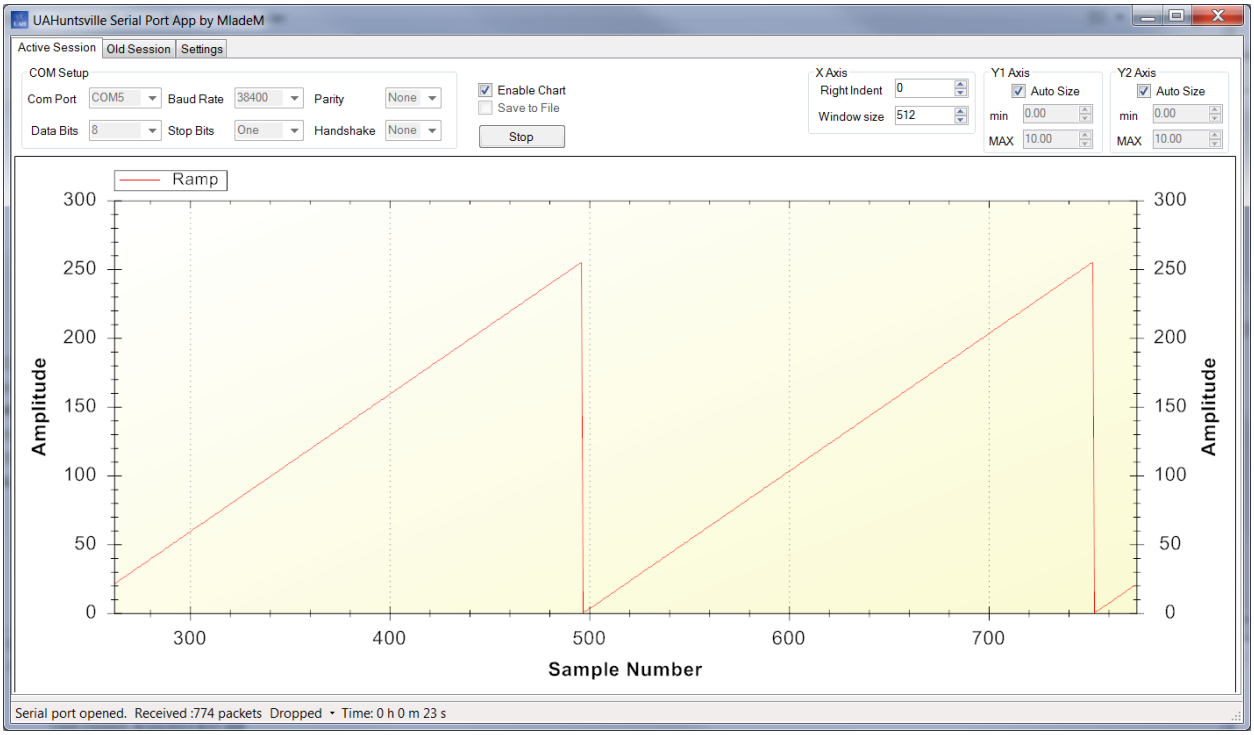

Figure 5. Chart screen.

1.3. Ramp_wave Program

Figure 6 shows the ramp_wave_floatconst program.

```
1. /*****************************************************************************
2. * Description: This program sends a 7-byte message via a UART channel.
3. *              It illustrates the use and configuration
4. *              of the UAH Serial Application.
5. *              Watchdog timer is configured in the interval mode.
6. *              It generates an interrupt request every 32 ms that wakes up CPU.
7. *              The main loop of the program sends a 7-byte message over UART:
8. *              8-bit header (0x55), followed by a 16-bit counter value,
9. *              and a 4-byte floating point value (constant set to 32768.0).
10.*              The counter is incremented every time the CPU is waken up.
11.*              The UAH Serial App should plot two signals, a ramp-like counter
12.*              and a constant set to 32,768.
13.*
14.* Platform:    TI Experimenter's board with MSP430FG4618/F2013
15.*
16.* Files:       ramp_wave_floatconst.c
17.*
18.* Setup:       Connect RS-232 cable from TI Experimenter's board with
19.*              MSP430FG4618/F2013 to a workstation that plots the.
20.*
21.*
22.* Author:      Aleksandar Milenkovic
23.* Contact:     milenkovic@computer.org
24.* Date:        June 2013
25.*****************************************************************************
   */
26.
27.#include <msp430xG46x.h>
28.
29.// UART Initializaion
30.void UART_Initialize(void);
31.
32.//send char function
33.void UART_putchar(char c);
34.
35.void main(void)
36.{
37.    unsigned int cnt = 0;
38.    const float hconst = 32768.0;
39.    char * p_cnt = (char *) & cnt;
40.    char * p_hconst = (char *) & hconst;
41.
42.    WDTCTL = WDT_MDLY_32; // watchdog timer in interval mode, 32 ms
43.    //Initialize the UART
44.    UART_Initialize();
45.    IE1 |= WDTIE;
46.    while (1) {
47.      _BIS_SR(LPM0_bits + GIE);
```

```
48.       cnt++;
49.       UART_putchar(0x55);
50.       //Send the integer value first, byte by byte
51.       UART_putchar(p_cnt[0]);
52.       UART_putchar(p_cnt[1]);
53.       //Send the float value, byte by byte
54.       UART_putchar(p_hconst[0]);
55.       UART_putchar(p_hconst[1]);
56.       UART_putchar(p_hconst[2]);
57.       UART_putchar(p_hconst[3]);
58.     }
59. }
60.
61. /* 38,400 bits/sec, 1 stop bit, no parity */
62. void UART_Initialize(void)
63. {
64.       P2SEL |= BIT4+BIT5; // Set UC0TXD and UC0RXD to transmit and receive data
65.       UCA0CTL1 |= BIT0;   // software reset
66.       UCA0CTL0 = 0;       // USCI_A0 control register
67.       UCA0CTL1 |= UCSSEL_2; // clock source SMCLK
68.       UCA0BR0=27;       // 1 MHz 38400
69.       UCA0BR1=0;        // 1 MHz 38400
70.       UCA0MCTL=0x94;    // Modulation
71.       UCA0CTL1 &= ~BIT0; // software resetIE2 |=UCA0RXIE;
72. }
73.
74. void UART_putchar(char c)
75. {
76.       // wait for other character to transmit
77.       while (!(IFG2 & UCA0TXIFG));
78.       UCA0TXBUF = c;
79. }
80.
81. // Watchdog Timer, interval mode, interrupt service routine
82. #pragma vector=WDT_VECTOR
83. __interrupt void watchdog_timer(void)
84. {
85.     _BIC_SR_IRQ(LPM0_bits); // Clear LPM0 bits from 0(SR)
86. }
87.
```

Figure 6. Ramp_wave_floatconst.c.

1.4. Configuration of UAH Serial App to Display Ramp Wave and Float Constant

Run the UAH Serial App and open up the settings tab as shown in Figure 7. Configure the parameters as follows.

- X-label: Sample Number (as we are not transmitting timestamps).
- Number of Samples to draw at the time: 1 (draw every new sample one by one)

- Number of Samples on graph = 2048
- Packet Size = 7 bytes
- Number of Channels = 2
- **CH 0 Configuration:**
    - Name: Ramp
    - Type: Byte 16bit (the payload to be plotted)
    - Position: 1 (header is byte 0, cnt is byte 1&2)
    - Show on Graph: checked
    - Y axis: Y1 checked
- **CH 1 Configuration:**
    - Name: FPConst
    - Type: Single 32bit (the payload to be plotted)
    - Position: 3 (header is byte 0, cnt is byte 1&2, fpconst is 3&4&5&6)
    - Show on Graph: checked
    - Y axis: Y1 checked

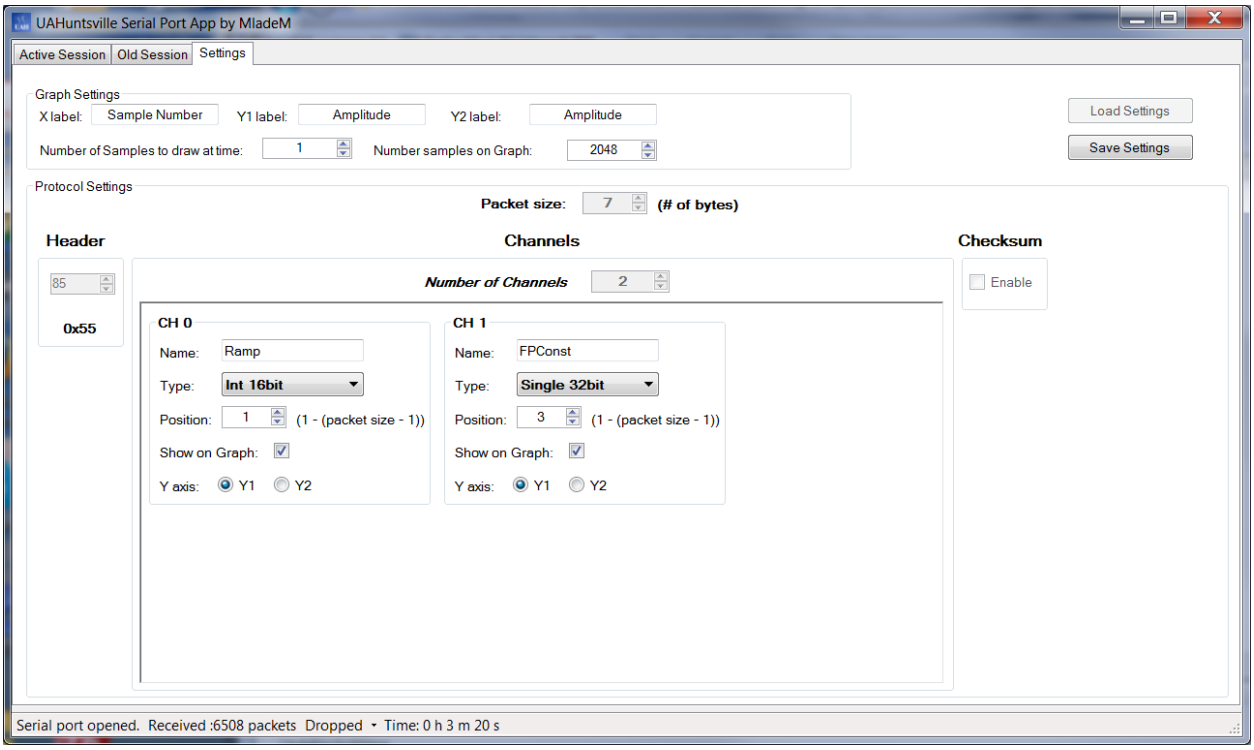

Figure 7. Settings screen configuration.

Figure 8 shows a chart screen for this example.

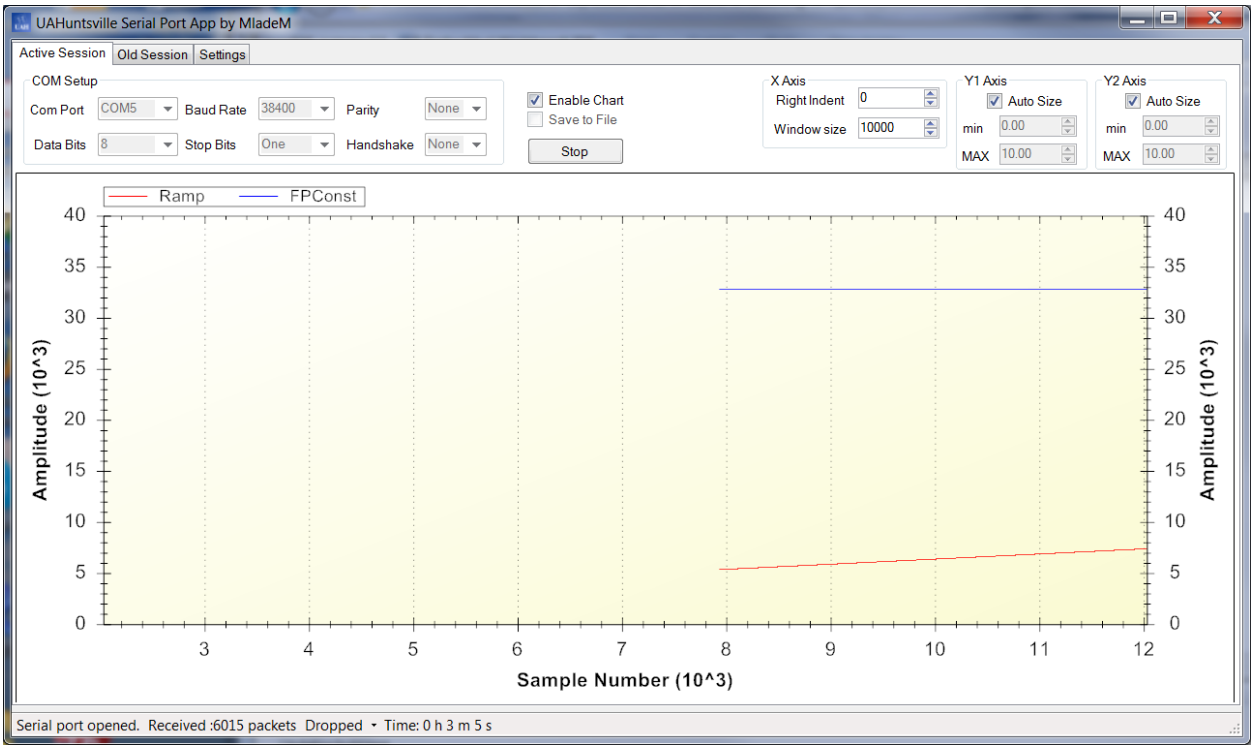

Figure 8. Chart screen.