CMU232

User Manual
Last Revised October 21, 2002
Overview

CMU232 is a new low-cost, low-power serial smart switch for serial data communications. It is intended for use by hobbyists to control multiple devices that use the RS232 protocol while only using one serial line. It essentially acts as a multiplexer for serial data, switching between three possible slave devices.

Capabilities

- RS232 Line Booster
- Baud Rate Converter
- TTL to RS232 Logic Converter
- Null Modem on Each Port
- 4 high-resolution general-purpose servo outputs and connectors, with selectable internal/external power source
- 50mhz SX Processor
- Serial Smart Switch
- Merge Mode
- 128k EEPROM
- Configurable status LEDs
- TTL line filtering

Updates

All updates and the latest news and revisions can be found online at http://www.cs.cmu.edu/~cmu232
Board Layout

!!!!!Finish Labelling this page
Assembled View
Ports

Power

The input power to the board goes through a 5-volt regulator. It is ideal to supply the board with between 6 and 7 volts of DC power that is capable of supplying at least 200mA of current.

Programming Port

The programming port allows the firmware to be downloaded to the SX28 using a SX-Key / Blitz or equivalent programmer. These can be purchased from Parallax Inc (www.parallaxinc.com). Most users will not need to use this option.
Servo Connectors

The CMU232 allows 4 servos to be controlled using either on board power or a separate power supply. Each connector, from SV1 to SV4 can be connected to one hobby servo, giving a total of 4 servos.

Servo Power:

JP2 controls servo power. To use the on board power supply, put a jumper at JP2.

You may also supply a separate power supply to the servos. This is recommended. To do this, connect GND and +5V from your separate power supply source as so.
Serial Ports

The CMU232 allows for four serial ports to be connected. One of these is the master port and the other three are referred to as slave ports.

Each Port is structured exactly the same way, and allows for both 5V TTL Logic and 12V Logic. There is also a null modem selector per each port.

Using 5V TTL Logic

To use 5V TTL Logic, connect the RX, TX and GND lines as so:

PC TX, Board RX  
GND  
PC RX, Board TX

You can easily null modem TTL Logic by switching the connector around (reversing TX and RX) since ground is the middle pin.
Using 12V Logic Serial Communications

First you must connect a jumper between terminals \( N_x \) and \( T_x \) to enable use of 12V Logic.

To enable normal operation, you will need to add 2 jumpers at NM\(_x\) and then connect the serial port. This port provides full level shifting for communication with a computer easily. Though it only uses 3 of the 10 pins it is packaged in a 2x5 configuration to fit standard 9 pin ribbon cable clip-on serial sockets and 10 pin female clip on serial headers that can both attach to a 10 wire ribbon cable.

If you wish to null modem the port, just change the direction of these two jumpers as so:
Finally to connect the 10 pin header to a 9 pin female serial ribbon cable head, connect as so:

The provided ribbon cable and connectors should make this an easy task.
Switches

Main Power

The main power switch is located here. The red LED right above it indicates that the board is powered.

![Main Power Switch Diagram]

Programming Switch

To change the board settings, the programming switch must be in the program position, marked by a ‘!’ on the board.

![Programming Switch Diagram]

For most operations the programming switch should remain in the normal operation mode position.

To change from programming mode to normal operation, the programming switch must be toggled, and the board must be reset by physically turning the board off and back on again. You will be able to tell whether the board is in programming mode by how many green status LEDs are on. If all the green status LEDs are on, then it is in programming mode.
Setting up the board

If your board has already been programmed, you can turn it on and you will see some of the green status port LEDs light up as well as the main power red LED light up. If all the green LEDs are on, you are in programming mode. If you wish to get out of programming mode, move the programming switch to the other position and reset the board.

If your board has not been programmed, you must use the programming port described earlier and an SXKey to flash the firmware onto the chip.

You must next download default settings for the board to operate. To do you must download the CMU232 Settings Programmer available at the CMU232 website at http://www.cs.cmu.edu/~cmu232

Follow these steps to first load the settings onto your CMU232 board:

1. Put the programming switch into programming mode and turn on the board. All the green LEDs should be lit.
2. Open the CMU232 Settings Programmer program.
3. Connect your computer COM port to the Master Serial Port of the CMU232.
4. Click “Open COM Port”
5. Click “Load Current Settings” if you wish to see what settings are currently on our board.
6. Select the settings that you wish to use on the board and click on “Save Current Settings.” You may wish to click Load Current Settings afterward to check that your settings have correctly been entered onto the board’s memory.
CMU232 Runtime Commands

These commands apply only to the default Smart Switch mode of operation for CMU232.

Switching to Slave 1 / Slave 2 / Slave 3

To switch to each slave, the default character sequence is a ‘\’ followed by the number of the slave device that you wish to switch to.

For example to switch to slave 3, you would type into the master line:
\3

You can use custom switch characters if you wish by programming them using the CMU232 Settings programmer.

Sending a ‘\’ character

The CMU232 ignores any letter after the first switch character (default = ‘\’) if that letter isn’t a switch command or a servo command. However, if you just wanted to pass a ‘\’ from the master line to the slave, you would just have to duplex the ‘\’ character. Example, to send a ‘\’ you would type into the master line:
\\n
Setting Servo Duty Cycle

To set servo duty cycle, the command sequences involves sending the first switch character (default = ‘\’), the letter ‘s’ or ‘S’, the number of the servo that you wish to control (1 through 4), the duty cycle that you wish the servo to be at (0 to 255), and a carriage return character (‘\r’).

For example to set servo 3 to duty 145, you would type:
\s3145<enter>

You should not enter any spaces between characters. The CMU232 will ignore all input from the master until a carriage return character (‘\r’) is detected after the initial “s” string.
## Parts Listing

<table>
<thead>
<tr>
<th>Description</th>
<th>Quantity</th>
<th>Digikey</th>
<th>Price Each</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power In Pin</td>
<td>1</td>
<td>WM2700-ND</td>
<td>0.59</td>
<td>0.59</td>
</tr>
<tr>
<td>Dual Row Headers (2*28 at least)</td>
<td>1</td>
<td>929852-01-36-ND</td>
<td>4.62</td>
<td>4.62</td>
</tr>
<tr>
<td>Single Row Headers (37 pins at least)</td>
<td>1</td>
<td>MHB50K-ND</td>
<td>2.72</td>
<td>2.72</td>
</tr>
<tr>
<td>SPST Switches</td>
<td>2</td>
<td>EG1847-ND</td>
<td>1.11</td>
<td>2.22</td>
</tr>
<tr>
<td>Red Power LED</td>
<td>1</td>
<td>167-1125-ND</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>Green Status LEDs</td>
<td>1</td>
<td>467-1127-ND</td>
<td>0.17</td>
<td>0.17</td>
</tr>
<tr>
<td>330 Ohm for LEDs</td>
<td>1</td>
<td>5330QBK-ND</td>
<td>0.06</td>
<td>0.30</td>
</tr>
<tr>
<td>100 Ohm for Analog Filters</td>
<td>4</td>
<td>100QBK-ND</td>
<td>0.06</td>
<td>0.24</td>
</tr>
<tr>
<td>10 KOhm I2C Pullups</td>
<td>2</td>
<td>10KQBK-ND</td>
<td>0.06</td>
<td>0.12</td>
</tr>
<tr>
<td>4.7 Kohm for SX MCLR Pullup</td>
<td>1</td>
<td>14.7KQBK-ND</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>50mhz Resonator</td>
<td>1</td>
<td>1250-15060-ND</td>
<td>1.53</td>
<td>1.53</td>
</tr>
<tr>
<td>1uF Electrolytic Caps for Max232</td>
<td>10</td>
<td>P2105-ND</td>
<td>0.42</td>
<td>4.20</td>
</tr>
<tr>
<td>0.1uF Ceramic Cap for SX Bypass</td>
<td>1</td>
<td>P4923-ND</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>100uF Low ESR Electrolytic Cap for Servos</td>
<td>1</td>
<td>P10323-ND</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>10uF 35V Electrolytic Cap for Before Regulator</td>
<td>1</td>
<td>P5178-ND</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>100uF Low ESR Cap for After Regulator</td>
<td>1</td>
<td>P10323-ND</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>0.01 or 0.001uF Ceramic Caps for Filtering</td>
<td>4</td>
<td></td>
<td>0.50</td>
<td>2.00</td>
</tr>
<tr>
<td>5 Volt Regulator</td>
<td>1</td>
<td>LM2940CT-5.0-ND</td>
<td>1.65</td>
<td>1.65</td>
</tr>
<tr>
<td>16 pin 0.3&quot; DIP socket</td>
<td>2</td>
<td>AE8916-ND</td>
<td>0.36</td>
<td>0.72</td>
</tr>
<tr>
<td>8 pin 0.3&quot; DIP socket</td>
<td>1</td>
<td>AE8908-ND</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>28 pin 0.3&quot; DIP socket</td>
<td>1</td>
<td>AE8928-ND</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>MAX232 Level Shifter (DIP)</td>
<td>1</td>
<td>MAX232CPE-ND</td>
<td>3.31</td>
<td>3.31</td>
</tr>
<tr>
<td>Ubicom SX28 Microcontroller (DIP)</td>
<td>1</td>
<td>SX28AC75/DP-ND</td>
<td>5.18</td>
<td>5.18</td>
</tr>
<tr>
<td>Microchip 24LC01 EEPROM (DIP)</td>
<td>1</td>
<td>24LC01B/P-ND</td>
<td>0.41</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Total: 33.93
Full Schematic (also available online in higher resolution)
CMU232 Manual

Page 15