## Setup

The following steps must be done before any free user operation can be done.

1. Connect the SPARC with the computer using a USB-jack cable.
2. Open the serial monitor to be used (e.g. Docklight™). Identify the COM port, and apply the following settings:

| | | | |
|---|---|---|---|
| Baud Rate | 9600 | Data Bits | 8 |
| Parity | None | Stop Bits | 1 |

3. Plug the SPARC.
4. Turn the red switch on, located at the back of the SPARC.
5. The serial monitor should receive the *WELCOME TO SPARC* message.
6. The SPARC will calibrate automatically. Once it finishes, the following messages will be sent:

*W*

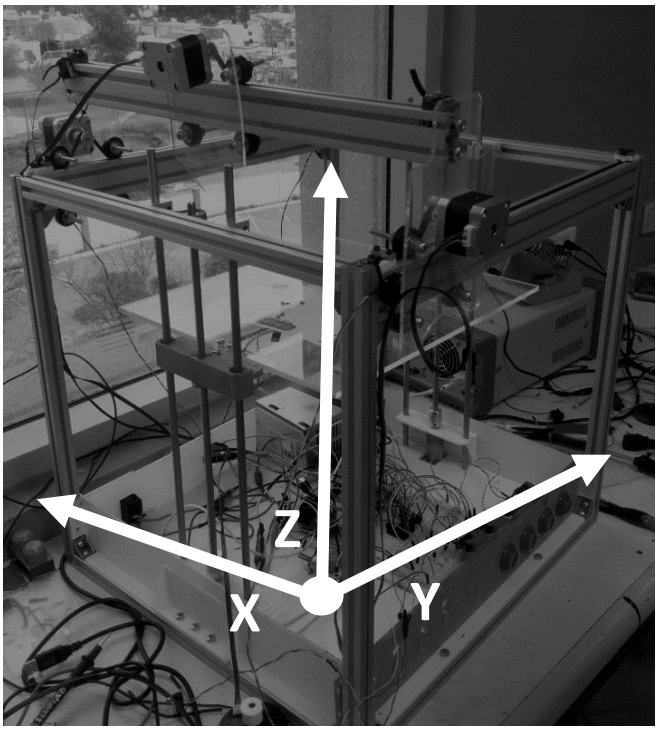
*C*

*R*

7. Send one of the three commands: **Memory**, **Origin** or **Adjust**.
8. Make sure the device is at the desired height.
9. Turn the black switch on, located at the side of the SPARC. The actuator will not work unless this switch is turned on.
10. The SPARC should now be set to work with the touchscreen.

## Coordinate system

The SPARC commands work under absolute coordinates as shown in the picture, the coordinate (000, 000, 000) is located at the junction.




## Commands

Before sending any command, make sure the serial monitor is in ASCII communication.

**Fast**

- Moves the actuator from the previous coordinates to the specified coordinates.
- The movement is performed with an approximate speed of XX mm/s.

Syntax:

| Opcode | Operand |
|---|---|
| F | HTO,HTO |

- 0 ≤ H ≤ 3 (Hundreds of millimeters.)
- 0 ≤ T ≤ 9 (Tenths of millimeters.)
- 0 ≤ O ≤ 9 (Ones of millimeters.)
- The first HTO refers to the X axis, the second HTO refers to the Y axis.

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- Upon completing the command, the microcontroller will send:

*C*

- If a coordinate out of physical range is introduced, the microcontroller will send:

*E1*

- If a coordinate contains non-numerical characters, the microcontroller will send:

*E2*

Example:

| Command | Expected behavior |
|---|---|
| F030,009 | 1. The message *W* appears in the serial monitor.<br>2. The actuator moves to the point located at 30 mm in the X axis and 9 mm in the Y axis.<br>3. The message *C* appears in the serial monitor. |

**Slow**

- Moves the actuator from the previous coordinates to the specified coordinates.
- The movement is performed with an approximate speed of XX mm/s.

Syntax:

| Opcode | Operand |
|---|---|
| S | HTO,HTO |

- $0 \leq H \leq 3$ (Hundreds of millimeters.)
- $0 \leq T \leq 9$ (Tenths of millimeters.)
- $0 \leq O \leq 9$ (Ones of millimeters.)
- The first HTO refers to the X axis, the second HTO refers to the Y axis.

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- Upon completing the command, the microcontroller will send:

*C*

- If a coordinate out of physical range is introduced, the microcontroller will send:

*E1*

- If a coordinate contains non-numerical characters, the microcontroller will send:

*E2*

Example:

| Command | Expected behavior |
|---|---|
| S030,009 | 1. The message *W* appears in the serial monitor.<br>2. The actuator moves to the point located at 30 mm in the X axis and 9 mm in the Y axis.<br>3. The message *C* appears in the serial monitor. |

**Touch**

- The actuator is activated, touching the screen for 100 ms, and then deactivated.

Syntax:

| Opcode | Operand |
|---|---|
| T | |

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- Upon completing the command, the microcontroller will send:

*C*

Example:

| Command | Expected behavior |
|---|---|
| T | 1. The message *W* appears in the serial monitor.<br>2. The actuator presses the touchscreen for 100 ms.<br>3. The message *C* appears in the serial monitor. |

**Hold**

- The actuator is activated; touching the screen.
- It remains activated until the Retract or Touch commands are sent.

Syntax:

| Opcode | Operand |
|---|---|
| H | |

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- Upon completing the command, the microcontroller will send:

*C*

Example:

| Command | Expected behavior |
|---|---|
| H | 1. The message *W* appears in the serial monitor.<br>2. The actuator extends, and remains extended.<br>3. The message *C* appears in the serial monitor. |

**Retract**

- The actuator is deactivated.
- It remains deactivated until the Retract or Touch commands are sent.

Syntax:

| Opcode | Operand |
|---|---|
| R | |

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- Upon completing the command, the microcontroller will send:

*C*

Example:

| Command | Expected behavior |
|---|---|
| R | 1. The message *W* appears in the serial monitor.<br>2. The actuator retracts, and remains retracted.<br>3. The message *C* appears in the serial monitor. |

**Origin**

- Load a set point stored in one of the EEPROM's 16 locations.
- The actuator is moved to the coordinates previously saved for the X and Y axes.
- The platform is set at the height specified for the Z axis if the value is within control limits.

Syntax:

| Opcode | Operand |
|---|---|
| O | D |

- 0 ≤ D ≤ F (Memory slot, in hexadecimal, from which the values will be loaded.)

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- After retrieving the data, the microcontroller will send:

*CDCXXXXCYYYYCZZZZ*

- Upon completing the command, the microcontroller will send:

*C*

- If a coordinate out of physical range is retrieved, the microcontroller will send:

*E1*

- If a retrieved coordinate contains non-numerical characters, the microcontroller will send:

*E2*

Example:

Assume the command MC010,015,155 was previously executed.

| Command | Expected behavior |
|---|---|
| OC | 1. The following message appears in the serial monitor:<br>*W*<br>*CC*<br>*CX 010*<br>*CY 015*<br>*CZ 155*<br>2. The platform moves to a 155 mm height.<br>3. The actuator moves to the point located at 10 mm in the X axis and 15 mm in the Y axis.<br>4. The message *C* appears in the serial monitor. |

**Memory**

- Store a set point in one of the EEPROM's 16 locations.

Syntax:

| Opcode | Operand |
|---|---|
| M | DHTO,HTO,HTO |

- 0 ≤ D ≤ F (Memory slot, in hexadecimal, in which the values will be stored.)
- 0 ≤ H ≤ 3 (Hundreds of millimeters.)
- 0 ≤ T ≤ 9 (Tenths of millimeters.)
- 0 ≤ O ≤ 9 (Ones of millimeters.)
- The first HTO refers to the X axis, the second HTO refers to the Y axis, and the third HTO refers to the Z axis.

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- After storing the data, the microcontroller will send:

*CD*

*CXS*

*CYS*

*CZS*

Example:

| Command | Expected behavior |
|---|---|
| MC010,015,155 | 1. The following message appears in the serial monitor:<br>*W*<br>*CC*<br>*CXS*<br>*CYS*<br>*CZS* |

**Adjust**

- The UP and DOWN buttons are enabled to adjust the platform.
- While Adjust is taking place, no commands will be executed.
- To exit Adjust, press the OK button.

Syntax:

| Opcode | Operand |
|---|---|
| A | |

Messages:

- Upon receiving the command, the microcontroller will send:

*W*

- If the OK button is pressed, the microcontroller will send:

*R*

Example:

| Command | Expected behavior |
|---|---|
| A | 1. The message *W* appears in the serial monitor.<br>2. If UP button is pressed, the platform rises. If DOWN button is pressed, the platform lowers.<br>3. When OK button is pressed, the message *R* appears in the serial monitor.<br>4. The UP and DOWN buttons will not perform any action when pressed. |

**End**

- End communication with the SPARC.

Syntax:

| Opcode | Operand |
|---|---|
| E | |

Messages:

- Upon receiving the command, the microcontroller will send:

*F*

Example:

| Command | Expected behavior |
|---|---|
| E | 1. The message *F* appears in the serial monitor.<br>2. The SPARC remains still.<br>3. No commands are executed by the SPARC. |

## Error messages

All error messages are listed below:

- The command sent was beyond the working space:

*E1*

- The command sent had characters that conflict with the syntax:

*E2*

- The command timed out. The incomplete command was discarded.

*E3*

- The command caused an overflow. The command was discarded.

*E4*