Robotics Systems Project Kit Manual

1 Introduction

This document provides a guide on how to setup the robotics kits, the required software, and describes how to use some of its basic functionality. Please read through this document before connecting any hardware.

1.1 List of Equipment

- 1 x Arduino Compatible Mega
- 1 x FE-URT-1 Serial to UART Board
- 1 x 7.5V 45W Power Supply & Cord
- 1 x 2.1mm DC Female to Leads
- 20 x M-F Jumper Leads
- 1 x Adafruit Analog 2-axis Joystick with Select Button
- 4 x FeeTech SCS15 16.5kg.cm Servo Motors + Accessories
- 2 x FeeTech SCS009 0.76kg.cm Servo Motors + Accessories

For more specific equipment details and motor specifications, please see the 'Robotics Systems Project Kit Specifications' document on CANVAS.

1.2 Software & Packages

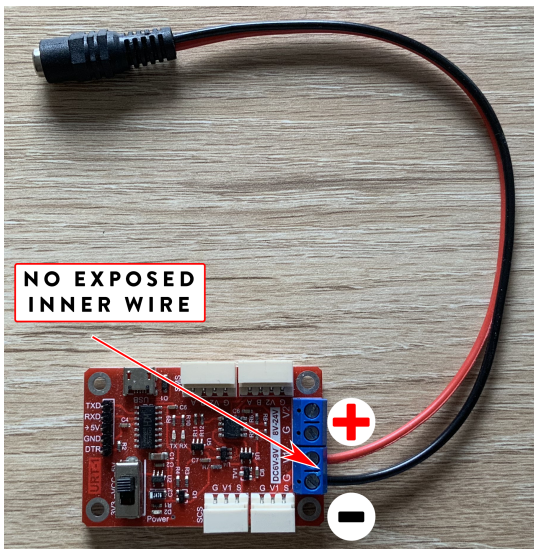
The provided skeleton code has been written to work with an Arduino Mega and MATLAB. In order for the Arduino to be able to communicate with the motors, the supplied SCS library is required. This can be downloaded from CANVAS along with the skeleton code. For details on how to install Arduino libraries, please refer to this guide: *www.arduino.cc/en/guide/libraries*.

While it is not required to use MATLAB, it is recommended to avoid heavy computation on the Arduino, and off-load it to capable computer-based software like MATLAB. The MATLAB skeleton code should not require any special libraries (although additional libraries may assist with the project), and has been written to optimise compatibility with older versions of MATLAB. As such, some notifications may be presented indicating that an approach that is not recommended"has been used, according to the R2021 guides. For further information, please refer to the guides available on the MathWorks website: *au.mathworks.com/help/index.html*

2 Assembling Equipment

2.1 Connecting FE-URT-1 To Power Cable

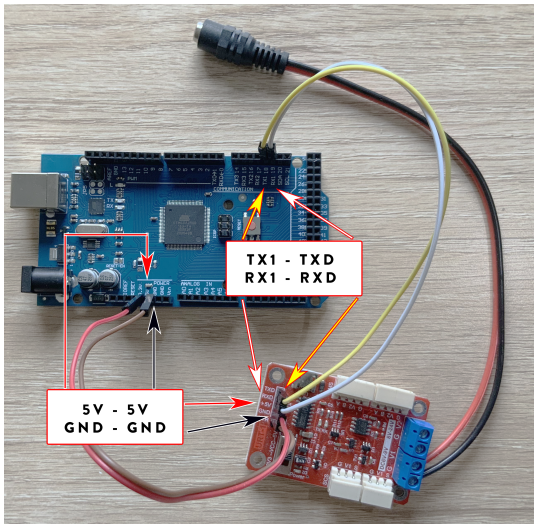
Connect the 2.1mm DC Female to Leads cable into the V1 and G terminals under the DC6V-9V block as shown in figure 1. Make sure that none of the internal wire of either lead is exposed and that the screw terminals are secured tightly.
NOTE: The power supply is connected to **THIS CABLE**, NOT THE ARDUINO!



Figur 1: FE-URT-1 Power Cable Assembly

2.2 Connecting FE-URT-1 to Arduino Mega

To connect the FE-URT-1 to the Arduino, firstly run cables between the 5V and GND pins of both boards. Then connect the TXD and RXD pins on the FE-URT-1 to the corresponding TX1 and RX1 pins on the Arduino Mega. See figure 2.



Figur 2: FE-URT-1 to Arduino Mega

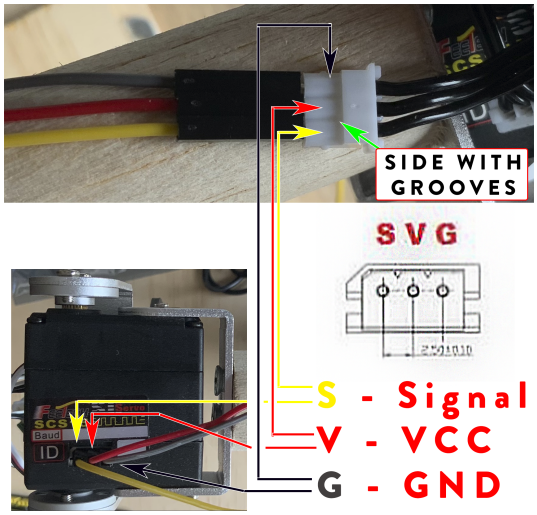
2.3 Connecting Motors to FE-URT-1

To connect the motors to the FE-URT-1 (and thus, the Arduino), plug the SCS motor cable into either port under the SCS header bank as shown in figure 3. **Note**: The motors can be daisy-chained together, and thus only one motore is required to be connected to the FE-URT-1. All subsequent motors should be connected into the previous one. It is also recommended that the motors be connected in ID order (i.e. ID-1 is connected to the FE-URT-1, ID-2 is connected to ID-1, etc...).

Figur 3: Extending the SCS Motor Cables

2.4 Extending the SCS Motor Cables

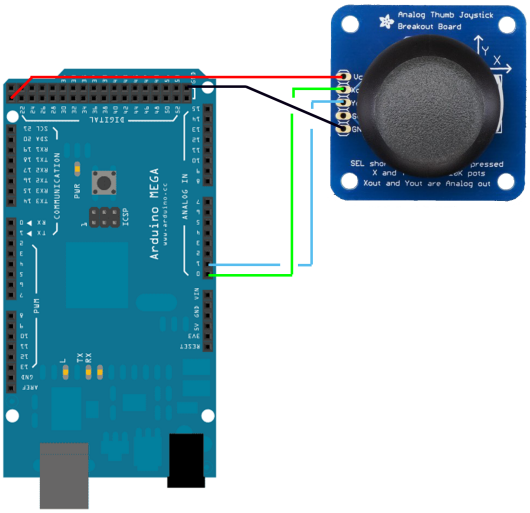
If an extension of the motor cables is required, this can be done using the provided jumper cables. Follow the diagram in figure 4 and ensure the Signal, VCC and GND cables are connected correctly.



Figur 4: Extending the SCS Motor Cables

2.5 Joystick

If the joystick is required, connect it to the Arduino Mega by using the 5V and GND pins located in the 'Digital' bank of pins on the Arduino. Then connect Xout and Yout from the joystick to the A0 and A1 pins on the Arduino respectively. See figure 5.



Figur 5: Extending the SCS Motor Cables

3 Controlling The Equipment

3.1 Skeleton Code

The skeleton code provided along with this manual sets up a serial connection between the Arduino and MATLAB, as well as giving some functions for elementary commands such as; receiving motor feedback, sending position and velocity commands, and changing the command mode.

3.1.1 Arduino

The Arduino skeleton code has mainly been written to handle the conversion of basic inputs into the required format of the FeeTech motors. It has been written in a way that can be build upon and expanded, and students can use the existing code in their assignments.

While the majority of basic functions have been accounted for, the list is by no means exhaustive, and it is encouraged that the provided code libraries and additional manuals are examined to gain a more complete understanding of how the motors function and what their limitations and capabilities are.

3.1.2 MATLAB

The MATLAB skeleton code consists of one primary live script along with several helper functions. The live script should be executed section-by-section and is only designed to provide guidance on how to perform basic operations. It is highly recommended that all helper functions are examined to get a thorough understanding of how the code works, and where necessary make modifications to suit the needs of the project.

3.2 Basic Commands

There are 5 basic commands implemented in the skeleton code, which should be enough for basic completion of the project. These are:

- Read Feedback - Returns a vector containing the position feedback of each motor in radians. The conversion from the SCS motor feedback format to radians is done in MATLAB, so if control is being done on the Arduino, a conversion will be required.
- Read Joystick - Returns x and y axis inputs from the joystick. This data is raw and may require processing such as calibration and filtering. This can be done in the readJoy.m file under 'Helper Functions'.
- Set Motor Command Mode - This sets the EPROM of all motors to either be in position or velocity control mode. Only execute this when a change of mode is required, NOT EVERY LOOP, as the EPROM only has a limited read/write lifespan.
- Send Position Command (Position Mode Only) - Send a vector of the desired angular positions to connected motors in radians. This will only work with the motor command mode set to position".
- Send Velocity Command (Velocity Mode Only) - Send a vector of the desired angular velocities to connected motors in rad/s. This will only work with the motor command mode set to "velocity".

NOTE: The index of the vectors corresponds to the connected motor ID (i.e. if all 6 motors are connected, the second element of the vector will correspond to the motor with ID-2. However, if ID-1 is not connected, but all others are, the second element in the vector will correspond to ID-3, as ID-2 will be element 1).

3.3 Advanced

For more advanced control and addressing of the hardware, please see the additional manuals located on CANVAS under 'Project Resources'. Some of these include; setting torque limits (useful for grippers), adjusting internal PID controllers (only affects position control mode) and reading addition feedback.

4 Technical Assistance & Error Reporting

If technical assistance is required regarding the setup of the kits or if a bug in the code is discovered, please post it on the CANVAS discussion board and a tutor will get back to you. Please read the discussion board before posting however, in case your issue has previously been resolved. Any updates to the skeleton code will be announced on CANVAS along with updated skeleton code.