# Arduino E.C. Sample Code

## Installation

1. **Download the Code**: Click the button below to download the `.ino` file.

## Usage

1. **Connect the System**: Ensure that your EC system is connected correctly. This includes connecting the power sources, data cables, and any necessary sensors or devices.
2. **Check the Configuration**: Confirm that all necessary parameters are configured correctly in your Arduino code.
3. **Upload the Code**: Transfer the code to your Arduino board using a USB cable or a matching communications tool.
4. **Test and Monitor**: After uploading, test the system to ensure it is functioning correctly and monitor the output data on the Serial Monitor of your Arduino IDE.

## Code Explanation

- **SoftwareSerial Setup**: Define how the soft serial port is going to work.
- **Function Definitions**: These functions are not enabled. Here are some functions you might find useful.
- **Serial Data Parsing**: The code uses `strtok` to parse the input data and extract specific values.
- **Serial Communication**: Use `Serial.println()` and other Serial commands to send and receive data.
- **Data Processing**: Parses the input data and performs calculations or operations based on the data received from the EC circuit.

## Additional Information

- **Baud Rate**: The old default baud rate was 38400. As of 11/6/14, the default baud rate has changed to 9600.
- **Data Format**: All data sent to the EC Circuit must end with a `<CR>` character. If the data from the EC circuit does not start with a number, we received. This will stop us from transmitting incorrect data. We add a 0 to the spot in the array just after the last character we received. We also count how many characters have been received. If we see that the EC circuit has sent a character, we read the data sent from the serial monitor until we see a `<CR>`. This will trigger the data received from the serial monitor. The interrupt will trigger when the data coming from the serial monitor has been received.
- **Data Reception**: If we see that the EC Circuit has sent a data, it means that we have new data from the EC Circuit. The code then adds the new data to the existing data in the array and compares it with the previous data. If the new data is not the same as the previous data, we mark the data as received. If the data from the EC circuit does not start with a number, we received. This will stop us from transmitting incorrect data. We add a 0 to the spot in the array just after the last character we received. We also count how many characters have been received. If we see that the EC circuit has sent a character, we read the data sent from the serial monitor until we see a `<CR>`. This will trigger the data received from the serial monitor. The interrupt will trigger when the data coming from the serial monitor has been received. If the data from the EC circuit does not start with a number, we received. This will stop us from transmitting incorrect data. We add a 0 to the spot in the array just after the last character we received. We also count how many characters have been received. If we see that the EC circuit has sent a character, we read the data sent from the serial monitor until we see a `<CR>`. This will trigger the data received from the serial monitor. The interrupt will trigger when the data coming from the serial monitor has been received. If the data from the EC circuit does not start with a number, we received. This will stop us from transmitting incorrect data. We add a 0 to the spot in the array just after the last character we received. We also count how many characters have been received. If we see that the EC circuit has sent a character, we read the data sent from the serial monitor until we see a `<CR>`. This will trigger the data received from the serial monitor. The interrupt will trigger when the data coming from the serial monitor has been received. If the data from the EC circuit does not start with a number, we received. This will stop us from transmitting incorrect data. We add a 0 to the spot in the array just after the last character we received. We also count how many characters have been received. If we see that the EC circuit has sent a character, we read the data sent from the serial monitor until we see a `<CR>`. This will trigger the data received from the serial monitor. The interrupt will trigger when the data coming from the serial monitor has been received.