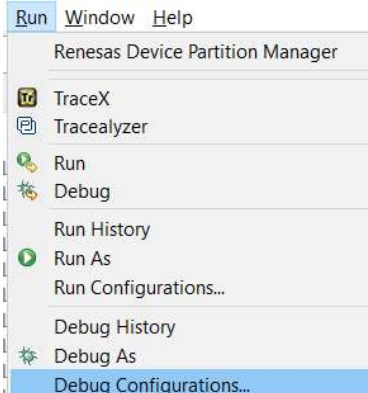How to Calibrate the Solar Charger Board

1) Start up e2studio and plug in the E2 Lite emulator to the PC USB as well as to the solar charger board.
2) Import "US011 Solar Charger FW.zip" and build the project. It should build with no errors and 2 warnings. Note the Tool Chain version is 1.10.
3) Select Run/Debug Configurations

4) Open up the Renesas GDB Hardware Debugging and select rl78g14_solar_battery_charger.x
   a. Click on the Debugger tab in the window on the right
   b. Go to the Connection Settings tab below that
   c. Make sure the Power Target From The Emulator is set to No
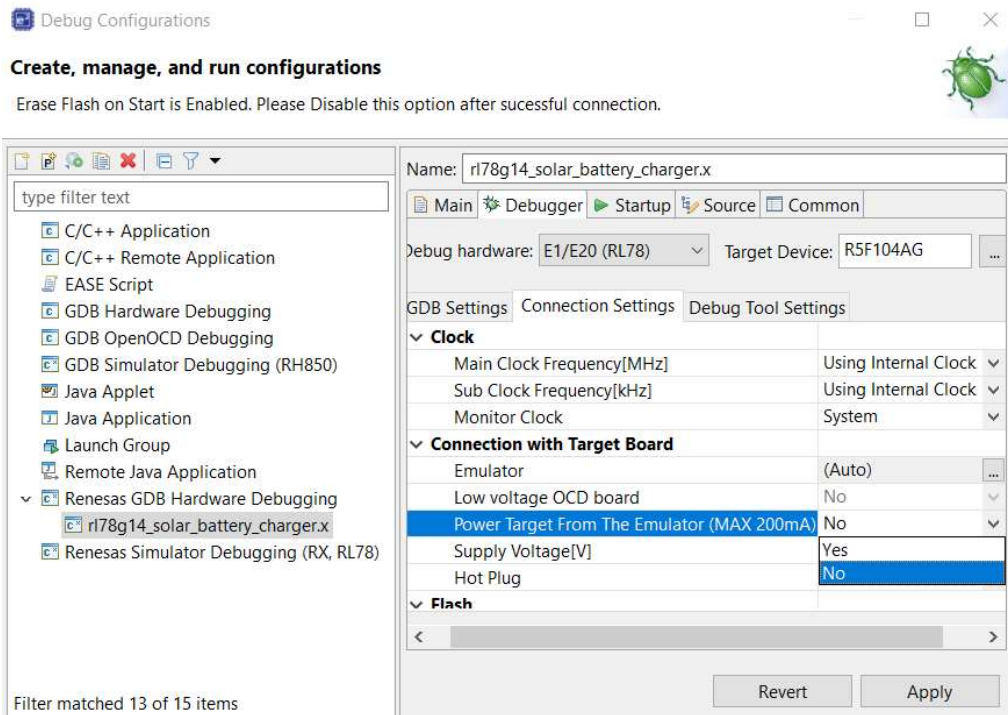
5) Connect input power to the board. The supply should be set to 18V and must be able to supply at least enough current to meet and exceed the desired float threshold. Turn on the power.
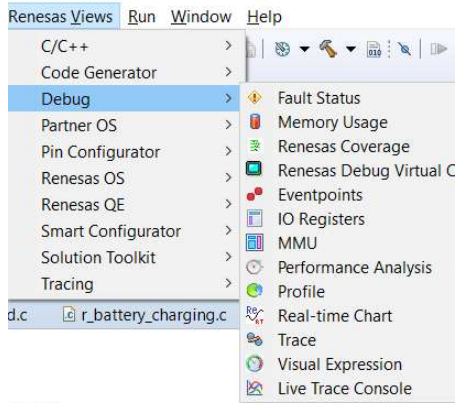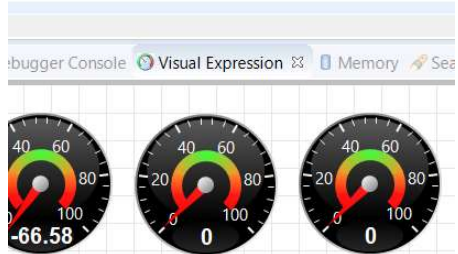
6) Load the firmware and debug the board.

Resume (F8)

7) Press the Resume button twice until the board is running

Running →

8) Go to the Visual Expressions tab. If you do not have the tab, find it under the Renesas View/Debug menu:

9) You should see the visual expressions dials read back the correct Vin, and Vout will activate.

15.64	**-6.84**	**13.24**	**2.74**
vin_ad	iin_ad	vout_ad	iout_ad
0	**0**	**23**	**3.42**
vin_iic	iin_iic	temperature_ad	mcu_voltage
15.64	**-7.02**	**13.24**	**2.74**
vin_filter	iin_filter	vout_filter	iout_filter

10) Read and record the iout_filter measurement from the ADC with no load. This is a more stable readout than the iout_ad.

11) Add a load equal to the desired float current threshold and record the iout_filter measurement again.

12) Power down Vin and then stop the debugger.

13) Set the threshold for iout reverse current (I suggest subtracting 0.01 or 0.02 from the no load value of iout_filter, depending on how much iout_ad jumps around) and enter it in r_main.c on line 140, after iout_ad <

```
*r_main.c
138                          r_sense_iin();
139                          r_sense_iout();
140                          if (vin_ad < 6 || vin_ad > 40 + 0.5 || vout_ad < 7 || iout_ad < 2.72)// ||
141                          {
```

14) Take the value for the float current threshold and enter it on line 71 of the r_userset.h for the IFLOAT_THRESHOLD

```
cstart.asm    r_main.c    *r_userset.h    r_board.c    r_b

64              #define MPPT_POWER_DELTA        0
65              #define MPPT_STEP_RATE_L        0.5
66              #define MPPT_STEP_RATE_S        0.2
67              #define MPPT_PEAK_COUNT         10
68              #define MPPT_STEP_CONVERT       120
69
70              /* battery charging */
71              #define IFLOAT_THRESHOLD        2.79
72              #define VBULK_THRESHOLD         14.5
```

15) On lines 156 and 171 of r_main.c, remove the **)//** to have the output only turn on when a battery is attached.

```
  *r_main.c ⊠   r_userset.h    r_board.c    r_mppt.h    r_mppt.c    r_perturb_and_observe.c    r_battery_charging.c
152
153        ⊖ 'val_100ms)
154
155          );
156        ⊖ .n_ad < 40 + 3.5 && temperature_ad > -30 && temperature_ad < 100)// && vout_ad > 8)
157
158          .on = 0;
159
160          )0ms = 0;
161
162
163        ⊖ ,
164
165        ⊖ 'val_900us)
166
167
168
169
170
171        ⊖ vin_ad < 40 && vout_ad < vout_set + 3.5 && iout_ad < 10 && iin_ad < 5)// && vout_ad > 7.5)
```

16) Build the project and Debug to load the corrected values into the MCU.
17) Stop the Debug and test the board on the bench/solar panel.