BLE Characteristics

This document contains information on the BLE Characteristics, their usage, UUIDs and some sample code.

Please note the full app was built to support both 5 and 10 output versions, however, the Apple Watch app should support 5 outputs only.

# SwitchMaster

SERVICE\_UUID

42574065-18ed-41e9-a723-b85ac1d74a42

## BLE CHAR 1

General Info

2ff2caa4-7932-44c6-b73a-a7cfb897d43c

*Open Tx for switches, voltage and instantaneous current*

Data Structure

uint8\_t ble\_adr[6]; // device mac

uint16\_t outputs; // output bitmask, LSB = Output1

uint16\_t voltage; // voltage in 10mV steps

uint16\_t current; // current in 10mA steps

Sample Code (Native)

Contained in ‘*Existing iOS Code*’ directory.

Sample Code (Flutter)

      List<int> smData = inComing.toList();

      int switchByte = smData[6];

      int switchByteB = smData[7];

      if (skipNextSwitch == 0) {

        for (var i = 1; i <= device.switchCount(); i++) {

          SwitchItem sw = device.getSwitch(i.toString());

          sw.isOn = (switchByte & (1 << i - 1) != 0);

        }

        if (device.switchCount() > 8) {

          for (var i = 1; i <= 2; i++) {

            SwitchItem sw = device.getSwitch((8 + i).toString());

            sw.isOn = (switchByteB & (1 << i - 1) != 0);

          }

        }

      } else {

        dev.log("Switch Status Skipped:" + skipNextSwitch.toString());

        skipNextSwitch = max(0, skipNextSwitch - 1);

      }

      // Get Volts.

      int vLow = smData[8];

      int vHigh = smData[9];

      vHigh = vHigh << 8;

      device.deviceVoltage = ((vLow + vHigh) / 100.0);

      int aLow = smData[10];

      int aHigh = smData[11];

      aHigh = aHigh << 8;

      device.totalDraw = ((aLow + aHigh) / 100.0);

## BLE CHAR 2

General Info

2ff2caa4-7932-44c6-b73a-a7cfb897d43d

*Secure channel for switching outputs*

Data Structure

uint8\_t ble\_adr[6]; // device mac// device mac

uint16\_t setting; // bitmask with settings

Sample Code (Native)

Contained in ‘*Existing iOS Code*’ directory.

Sample Code (Flutter/Dart)

List<int> payload = [];

    int switchByte = 0;

    int switchByteB = 0;

    for (var i = 1; i <= device.switchCount(); i++) {

      SwitchItem sw = device.getSwitch(i.toString());

      if (sw.isOn) {

        switchByte = switchByte | (1 << (i - 1));

      }

    }

    if (device.switchCount() > 8) {

      for (var i = 1; i <= 2; i++) {

        SwitchItem sw = device.getSwitch((8 + i).toString());

        if (sw.isOn) {

          switchByteB = switchByteB | (1 << (i - 1));

        }

      }

    }

    List<int> switchArray = [switchByte, switchByteB];

    Uint8List switchArrayBytes = Uint8List.fromList(switchArray);

    payload.addAll(await \_encodeMac(device));

    payload.addAll(switchArrayBytes);

    while (payload.length < 8) {

      payload.add(0);

    }

## BLE CHAR 3

General Info

2ff2caa4-7932-44c6-b73a-a7cfb897d46b

*Open Tx for channel lockouts*

Data Structure

uint8\_t ble\_adr[6]; // device mac

uint16\_t ch\_lockout[MAX\_CHANNELS]; // individual lockout value - 0 off. 1 locked

Sample Code (Flutter/Dart)

        for (var i = 1; i <= device.switchCount(); i++) {

          SwitchItem sw = device.getSwitch(i.toString());

          int offSet = 0;

          if (i == 1) {

            offSet = 0;

          } else if (i > 1) {

            offSet = device.switches.indexOf(sw) \* 2;

          }

          int lockByteA = lockoutdata[offSet + 6];

          if (lockByteA == 1) {

            sw.isLocked = true;

          } else {

            sw.isLocked = false;

          }

        }

## BLE CHAR 4

General Info

2ff2caa4-7932-44c6-b73a-a7cfb897d440

*Open Tx for current averages*

Data Structure

uint8\_t ble\_adr[6]; // device mac

uint16\_t last\_hour; // average current the last hour in 10mA steps

uint16\_t last\_12; // average current the last 12 hours in 10mA steps

Sample Code (Flutter/Native)

      List<int> smData = inComing.toList();

      int cLow = smData[6];

      int cHigh = smData[7];

      cHigh = cHigh << 8;

      device.current\_1hr = ((cLow + cHigh) / 100.0);

      cLow = smData[8];

      cHigh = smData[9];

      cHigh = cHigh << 8;

      device.current\_12hr = ((cLow + cHigh) / 100.0);