Chessnut chess board communications

by Graham O'Neill

1. Introduction

The board connects to the PC using either a standard USB cable which it uses for both signals and power supply or using BluetoothLE. The USB connection uses HID technology rather than serial port communications.

It has 64 RFID scanners built into it. Each piece has an RFID chip built into it that the board uses to identify the piece and translate it into the appropriate piece code. Each square has a single LED on which can be on or off (i.e. the hardware has no mechanism to flash the LEDs - if required that has to be done from the software).

2. Communications using USB

For HID communications you need to discover HID devices first since the Product ID can vary depending on the batch of manufacture. You need to connect to the device with the correct UsagePage (0xFF00) as there is also one that uses 0x0001.

Product Name:	ChessAir
Vendor ID:	0x2D80
Product ID:	0x8001 or 0x8002 or...
Write report ID:	0x21 and 0x0A
Read report ID:	0x01
Usage page:	0xFF00

2.1 Communications from the board

The signals from the board consist of a sequence of 65 bytes. The first two are:

```
0x01 0x3D
```

which indicate that this is a position signal. The next 32 bytes specify the position and I don't currently know what the following 31 bytes are used for.

Each square has a value specifying the piece:

Value	0	1	2	3	4	5	6	7	8	9	A	B	C
Piece	.	q	k	b	p	n	R	P	r	B	N	Q	K

Each of the 32 bytes represents two squares with the order being the squares labelled H8,G8,F8...C1,B1,A1. Within each byte the lower 4 bits represent the first square and the higher 4 bits represent the second square. This means that if the 32 bits were written out in normal hex characters the pairs would actually appear reversed.

For example, the 32 bytes for the normal starting position with black on the 7th and 8th ranks would be shown as:

```
58 23 31 85 44 44 44 44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 77 77 77 77 A6 C9 9B 6A
```

So the first byte's value of 0x58 means a black rook (0x8) on H8 and a black knight (0x5) on G8.

2.2 Communications to the board

When the board is first connected it is necessary to send it a three byte initialisation code:

```
0x21 0x01 0x00
```

After that the position data will be sent from the board every 200ms or so.

The other data sent to the board controls the LEDs. There are two control bytes and 8 data bytes:

```
0x0A 0x08 <R8> <R7> <R6> <R5> <R4> <R3> <R2> <R1>
```

where the 8 bytes represent the LEDs with one byte for each row of the board. The first byte is for the row furthest away (labelled A8..H8).

For each byte the value is determined by whether the LED for each square needs to be on or off. If the square is off then it will have a value of 0 and if it needs to be on then the value will be based on the square position in the row, with values being:

```
128 64 32 16 8 4 2 1
```

The values for all the squares in the row are added together, meaning the maximum value of the byte is 255 which would occur if all of the LEDs in the row were turned on. So to show the move E2-E4 (with the board in the normal, non-flipped position) the ten bytes (including the controls) would be:

```
0A 08 00 00 00 00 08 00 08 00
```

To turn off all LEDs you just send the 10 bytes with the last 8 bytes all as zero values.

3. Communications using BLE

For BLE the details are:

Device name:	Chessnut Air (or Smart Chess)
Write characteristic:	1B7E8272-2877-41C3-B46E-CF057C562023
Read (board data):	1B7E8262-2877-41C3-B46E-CF057C562023
Read (confirmation):	1B7E8273-2877-41C3-B46E-CF057C562023

The two Read characteristics are both Notify types but note they are under different Services (1B7E8261... and 1B7E8271...).

2.1 Communications from the board

The signals from the board consist of a sequence of 38 bytes. The first two are:

```
0x01 0x24
```

which indicate that this is a position signal. The next 32 bytes specify the position in the same way as the USB data and I don't currently know what the following 4 bytes are used for.

2.2 Communications to the board

When the board is first connected it is necessary to send it a three byte initialisation code:

```
0x21 0x01 0x00
```

The board will reply with three bytes to the confirmation characteristic:

```
0x23 0x01 0x00
```

After that the position data will be sent from the board to the board data characteristic every 200ms or so. Note that this is different to other BLE boards which often only send data when a piece is moved.

The other data sent to the board controls the LEDs, which is done in exactly the same way as the USB connection. After sending an LED command of 10 bytes the board will reply with the same three byte confirmation code.