0616

（前回のミーティング）

・表示フィルターに ip.dstAddr == “127.0.0.1” && port.dst == 423 のようにフィルターをかける. Protocol に IEEE802.15.4 が表示されるはずである.

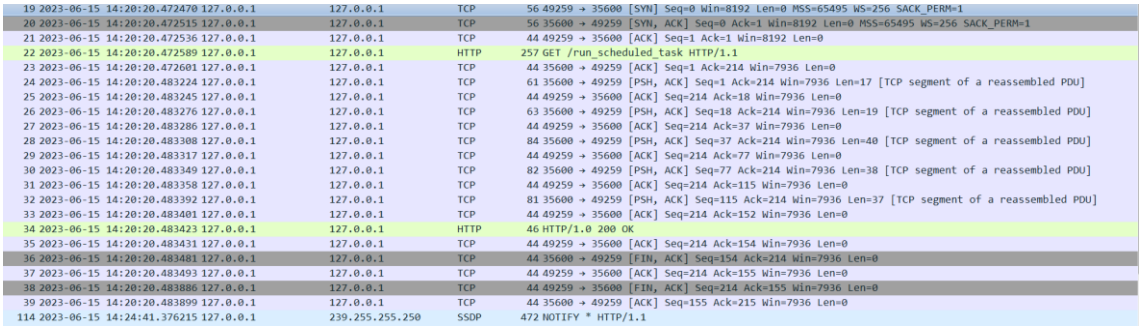
・本研究では, Raspberry Pi を使用することに決定した.

【今週の進捗】

127.0.0.1 は IP アドレスで自分自身のことを指す. 以下は今回試したことである.

1．Source に 127.0.0.1 は確認できたが, Protocol が IEEE802.15.4 ではなく, SSDP であった.SSDP(Simple Service Discovery Protocol)が生成しているもので, SSDP はプラグアンドプレイデバイス検出に使用されているプロトコルである.

2．再度, 127.0.0.1 のみにフィルターをかけて，行った.



19	2023-06-15 14:20:20.472470	127.0.0.1	127.0.0.1	TCP	56 49259 → 35600 [SYN] Seq=0 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
20	2023-06-15 14:20:20.472515	127.0.0.1	127.0.0.1	TCP	56 35600 → 49259 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=65495 WS=256 SACK_PERM=1
21	2023-06-15 14:20:20.472536	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=1 Ack=1 Win=8192 Len=0
22	2023-06-15 14:20:20.472589	127.0.0.1	127.0.0.1	HTTP	257 GET /run_scheduled_task HTTP/1.1
23	2023-06-15 14:20:20.472601	127.0.0.1	127.0.0.1	TCP	44 35600 → 49259 [ACK] Seq=1 Ack=214 Win=7936 Len=0
24	2023-06-15 14:20:20.483224	127.0.0.1	127.0.0.1	TCP	61 35600 → 49259 [PSH, ACK] Seq=1 Ack=214 Win=7936 Len=17 [TCP segment of a reassembled PDU]
25	2023-06-15 14:20:20.483245	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=18 Win=7936 Len=0
26	2023-06-15 14:20:20.483276	127.0.0.1	127.0.0.1	TCP	63 35600 → 49259 [PSH, ACK] Seq=18 Ack=214 Win=7936 Len=19 [TCP segment of a reassembled PDU]
27	2023-06-15 14:20:20.483286	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=37 Win=7936 Len=0
28	2023-06-15 14:20:20.483308	127.0.0.1	127.0.0.1	TCP	84 35600 → 49259 [PSH, ACK] Seq=37 Ack=214 Win=7936 Len=40 [TCP segment of a reassembled PDU]
29	2023-06-15 14:20:20.483317	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=77 Win=7936 Len=0
30	2023-06-15 14:20:20.483349	127.0.0.1	127.0.0.1	TCP	82 35600 → 49259 [PSH, ACK] Seq=77 Ack=214 Win=7936 Len=38 [TCP segment of a reassembled PDU]
31	2023-06-15 14:20:20.483358	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=115 Win=7936 Len=0
32	2023-06-15 14:20:20.483392	127.0.0.1	127.0.0.1	TCP	81 35600 → 49259 [PSH, ACK] Seq=115 Ack=214 Win=7936 Len=37 [TCP segment of a reassembled PDU]
33	2023-06-15 14:20:20.483401	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=152 Win=7936 Len=0
34	2023-06-15 14:20:20.483423	127.0.0.1	127.0.0.1	HTTP	46 HTTP/1.0 200 OK
35	2023-06-15 14:20:20.483431	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=154 Win=7936 Len=0
36	2023-06-15 14:20:20.483481	127.0.0.1	127.0.0.1	TCP	44 35600 → 49259 [FIN, ACK] Seq=154 Ack=214 Win=7936 Len=0
37	2023-06-15 14:20:20.483493	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [ACK] Seq=214 Ack=155 Win=7936 Len=0
38	2023-06-15 14:20:20.483886	127.0.0.1	127.0.0.1	TCP	44 49259 → 35600 [FIN, ACK] Seq=214 Ack=155 Win=7936 Len=0
39	2023-06-15 14:20:20.483899	127.0.0.1	127.0.0.1	TCP	44 35600 → 49259 [ACK] Seq=155 Ack=215 Win=7936 Len=0
114	2023-06-15 14:24:41.376215	127.0.0.1	239.255.255.250	SSDP	472 NOTIFY * HTTP/1.1

送信元も宛先も 127.0.0.1 である．プロトコルが IEEE ではなく TCP であった.

3．次に宛先である Coordinator の 64bit の Mac アドレス(0x001BC50122016BD5)でも以下のように，フィルターかけることが可能であったため，小文字表記で条件を絞る.

・eth.addr == 1b:c5:01:22:01:6b:d5

・eth.dst ==1b:c5:01:22:01:6b:d5

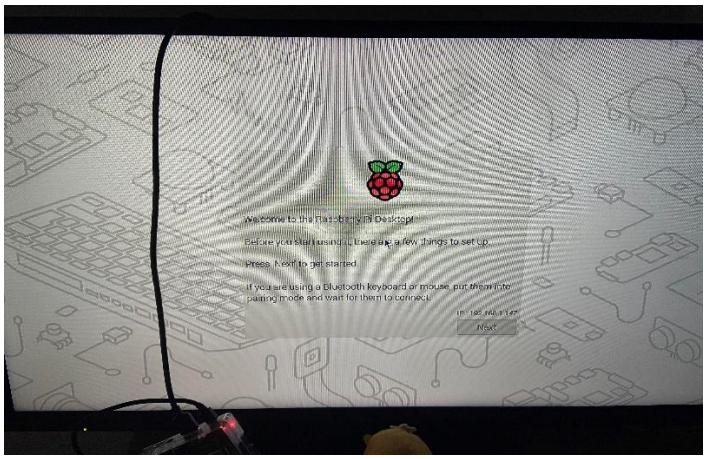
しかし，キャッチできず，何も表示されなかった.

【ラズパイの初期設定】

1. ラズパイに OS を入れるために、Raspberry Pi Imager をダウンロード/インストールを行った.
2. Raspberry Pi Imager を用いて、Raspberry Pi OS（32-bit）をラズパイの SD カードにインストールした.
3. モニターとラズパイを接続し、ラズパイの初期設定を行うことで、使用することが可能になる.

・今回はラズパイの OS のインストールを自分のパソコン(Windows)で行いたいため、Windows 用の Raspberry Pi Imager を使用した.

・Arduino とラズパイを同じものだと判断して作業をしていたが，ラズパイは独立したコンピューターであるため，ノート PC のモニターではラズパイを表示することはできない.
以下の写真のように，一時的にラズパイとテレビを繋いで, OS の有無を確認した.



・ラズパイにターミナルがあるため、そこでプログラムのコンパイル等を行う.
AD 変換に関しては, D++か Python でのプログラミングとなる. 今のところ Python でのプログラミングを予定しているが，相談して変更する可能性がある.

【スケジュール】

	スケジュール	実施したこと	できなかったこと	来週への課題
5/26 ~ 6/2	・JN5169にbeaconがないため JN5189を使用	・JN5189を検討結果使用しない ・pollコードを制御	・wiresharkの全般の理解	・JN5169を継続 ・E→Cの送信で検証 ・wiresharkでの確認
6/2 ~ 9	・E→Cでの送信を wiresharkで確認	・E→Cでの送信	・wiresharkでの正確な表示	・ArduinoかRaspberry Piを用いてAD変換を実施する.
6/9 ~ 16	・ラズパイの初期設定 ・フィルタありのwireshark	・ラズパイの初期設定 ・wiresharkのフィルタで表示内容を制限	・適切なフィルター表示	・wiresharkで指定の パケットのみを表示 ・ラズパイでデータ収集 ・AD変換のプログラミング (Pythonの予定)
6/16 ~7/7	・実際にセンサを使用し, UARTを設定			