

M5Stack-TimerCameraで 定期・定刻のGoogleDrive画像保存

- NTPによる時刻同期及び、定期・定刻起動
- GoogleAPIを用いたGoogleDrive保存

目次

1. 機器選定
2. 開発環境
3. ArduinoIDE設定
4. NTPについて
5. Google設定 (Refreshトークンの取得など)
6. Arduinoプログラム
7. プログラム書き込み
8. 動作確認

過去の動画

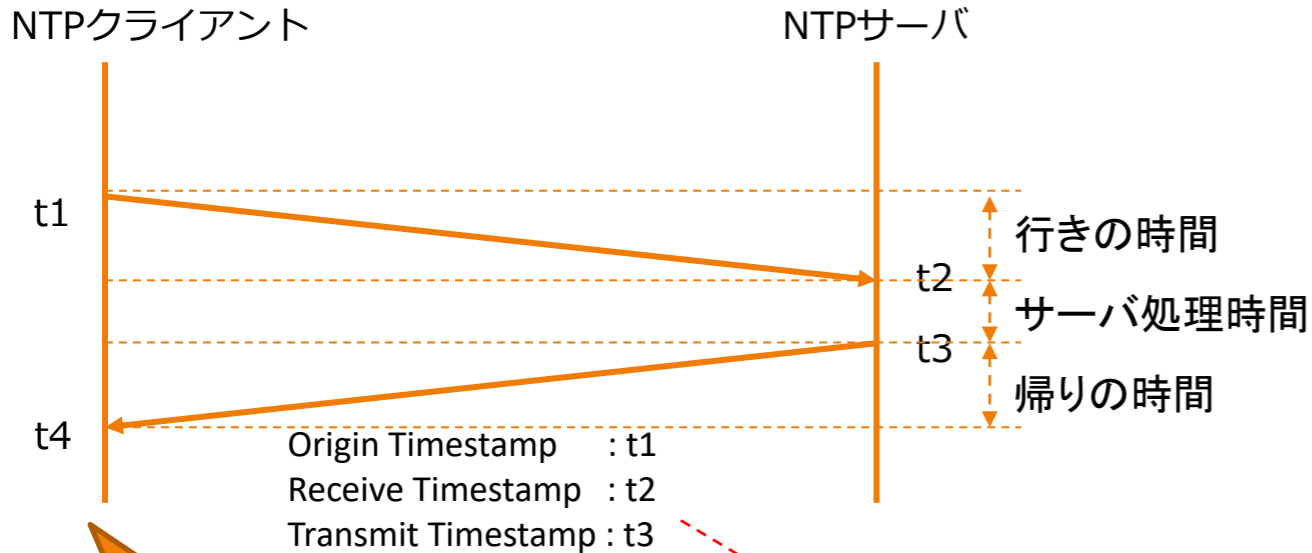
《スマホで動画視聴》
https://hobby-it.com/m5timer_webcam

《GoogleクラウドAPI利用のための
設定を徹底解説》
<https://hobby-it.com/googledriveapi/>

《ESP32とOV2640でGoogleDrive保存【API】
【ソフトウェア(プログラミング)編】》
<https://hobby-it.com/save-jpeg-image-with-gdriveapi-3/>

4. NTPについて

●NTP: Network Time Protocol
ネットワーク機器が自動で時刻を設定するプロトコル



$$\text{Offset(時間差)} = \frac{\text{行きの時間} \quad \text{帰りの時間}}{2} = \frac{(t2-t1) - (t4-t3)}{2}$$

●Windows11端末で時刻同期

1) 行きパケット

ntp_capture.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.9	20.43.94.199	NTP	90	NTP Version 3, client
2	0.023148	20.43.94.199	192.168.1.9	NTP	90	NTP Version 3, server

Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF...
Ethernet II, Src: Tp-LinkT_09:d6:7d (28:ee:52:09:d6:7d), Dst: Mitsubis_86:d6:65 (10:4b:46:86:d6:65)
Internet Protocol Version 4, Src: 192.168.1.9, Dst: 20.43.94.199
User Datagram Protocol, Src Port: 123, Dst Port: 123
Network Time Protocol (NTP Version 3, client)
Flags: 0xdb, Leap Indicator: unknown (clock unsynchronized), Version number: NTP
[Response In: 2]
Peer Clock Stratum: unspecified or invalid (0)
Peer Polling Interval: 17 (131072 seconds)
Peer Clock Precision: 0.000000 seconds
Root Delay: 0.000000 seconds
Root Dispersion: 1.000000 seconds
Reference ID: NULL
Reference Timestamp: Jan 12, 2023 00:17:57.812058499 UTC
Origin Timestamp: NULL
Receive Timestamp: NULL
Transmit Timestamp: Feb 9, 2023 08:58:29.733061599 UTC

2) 帰りパケット

ntp_capture.pcapng

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.9	20.43.94.199	NTP	90	NTP Version 3, client
2	0.023148	20.43.94.199	192.168.1.9	NTP	90	NTP Version 3, server


Frame 2: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF...
Ethernet II, Src: Mitsubis_86:d6:65 (10:4b:46:86:d6:65), Dst: Tp-LinkT_09:d6:7d (28:ee:52:09:d6:7d)
Internet Protocol Version 4, Src: 20.43.94.199, Dst: 192.168.1.9
User Datagram Protocol, Src Port: 123, Dst Port: 123
Network Time Protocol (NTP Version 3, server)
Flags: 0xc1c, Leap Indicator: no warning, Version number: NTP Version 3, Mode: set
[Request In: 1]
[Delta Time: 0.023148000 seconds]
Peer Clock Stratum: secondary reference (3)
Peer Polling Interval: 17 (131072 seconds)
Peer Clock Precision: 0.000000 seconds
Root Delay: 0.002106 seconds
Root Dispersion: 0.029526 seconds
Reference ID: 25.00.230.0
Reference Timestamp: Feb 9, 2023 08:48:56.564738399 UTC
Origin Timestamp: Feb 9, 2023 08:58:29.733061599 UTC
Receive Timestamp: Feb 9, 2023 08:58:29.954736299 UTC
Transmit Timestamp: Feb 9, 2023 08:58:29.954739399 UTC

4. NTPについて

1)は時間差なし。2)~4)は20秒遅れてる場合


1)時間差なし、行き帰り15秒

t1) 13:00 } 行き:15秒
 t2) 13:15 } (サーバ処理:5秒)
 t3) 13:20 } 帰り:15秒
 t4) 13:35 }

$$\frac{15 - 15}{2} = 0$$



2)行き帰りが15秒の場合

t1) 13:00 } 行き:15秒
 t2) 13:35 } (サーバ処理:5秒)
 t3) 13:40 } 帰り:15秒
 t4) 13:35 }

$$\frac{35 - (-5)}{2} = 20$$



3)行きが5秒小さい場合

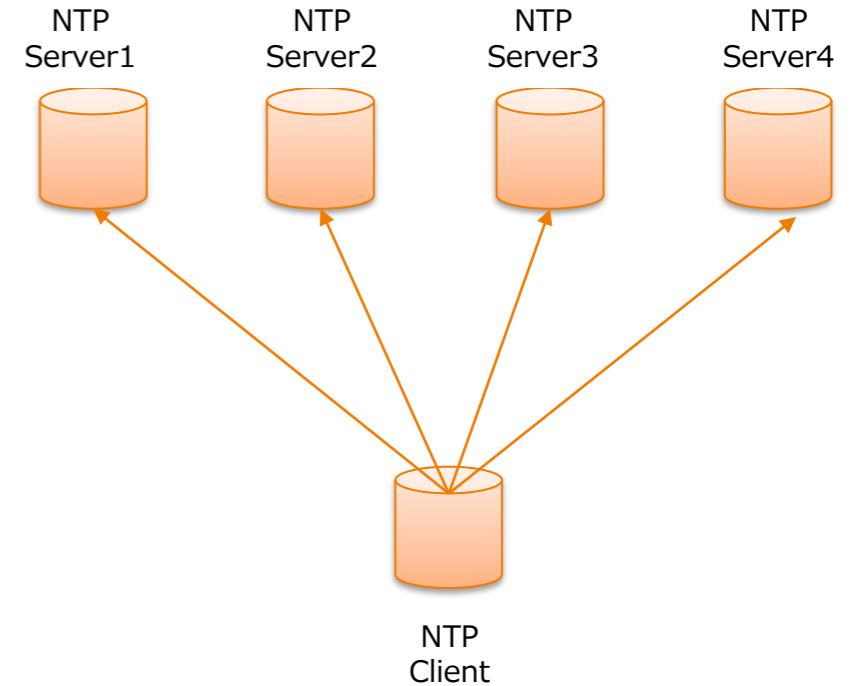
t1) 13:00 } 行き:10秒
 t2) 13:30 } (サーバ処理:5秒)
 t3) 13:35 } 帰り:15秒
 t4) 13:30 }

$$\frac{30 - (-5)}{2} = 17.5$$


4)帰りが5秒小さい場合

t1) 13:00 } 行き:15秒
 t2) 13:35 } (サーバ処理:5秒)
 t3) 13:40 } 帰り:10秒
 t4) 13:30 }

$$\frac{35 - (-10)}{2} = 22.5$$




3つの信頼できるサーバから時刻同期をおこなうことで精度を向上

NTPは通信時間が行きと帰りで同じことが前提

6. Arduinoプログラム

```
17 // #####_Line_Wi-Fi_settings_(Preferences)#####
18 String clientId      = "##### CLIENT-ID #####.apps.googleusercontent.com"; // $$$ CHANGE REQUIRED $$$
19 String clientSecret  = "##### CLIENT-SECRET #####"; // $$$ CHANGE REQUIRED $$$
20 String refreshToken  = "##### REFRESH-TOKEN #####"; // $$$ CHANGE REQUIRED $$$
21 String driveFolder  = "##### GOOGLE-FOLDER-ID #####"; // $$$ CHANGE REQUIRED $$$
22
23 const char *ssid     = "##### SSID #####"; // $$$ CHANGE REQUIRED $$$
24 const char *password = "### PASSWORD ###"; // $$$ CHANGE REQUIRED $$$
25
26 const int Interval  = 20; // Image save interval (minutes) [OFF:-1]
27 const int SavaTime  = 15; // Image save time (0 to 24 hours exactly) [OFF:-1]
28 // #####
29 const char* refreshServer = "oauth2.googleapis.com";
30 const char* refreshUri    = "/token";
31 const char* apiServer    = "www.googleapis.com";
32 const char* apiUri       = "/upload/drive/v3/files?uploadType=multipart";
33 String accessToken      = "";
34
35 int preMin = -1; // last run time(for Interval)
36 int preHour = -1; // last run time(for Time)
37
38 // LED Pin Setting
39 const byte LED_PIN = 2; // Green LED
..
```

Google及びWi-Fi設定

必ず変更必要

定期起動及び定刻起動設定

前回の定期起動及び定刻起動時間を
保存するための変数

```
141
142 // ##### NTP setting #####
143 configTime(9 * 3600L, 0, "ntp.nict.jp", "time.google.com", "ntp.jst.mfeed.ad.jp");
144
```

NTPの設定
(NTPサーバを設定)

6. Arduinoプログラム

```
148 // Loop Function
149 void loop() {
150     struct tm timeInfoLoop;
151     getLocalTime(&timeInfoLoop);
152     bool doFlag = false;
153     // Check Interval
154     int curMin = -1;
155     if ( preMin <= timeInfoLoop.tm_min ) {
156         curMin = timeInfoLoop.tm_min;
157         // When time advances beyond 60 minute
158     } else {
159         curMin = timeInfoLoop.tm_min + 60;
160     }
161     // Check Interval time
162     if ( Interval > 0 && ( (curMin - preMin) >= Interval || preMin == -1 ) ) {
163         doFlag = true;
164         preMin = timeInfoLoop.tm_min;
165         Serial.print("Check Interval: ");
166         Serial.println(preMin);
167     }
168     // Check Time
169     if ( preHour != timeInfoLoop.tm_hour && SaveTime == timeInfoLoop.tm_hour && timeInfoLoop.tm_min == 0 ) {
170         doFlag = true;
171         preHour = timeInfoLoop.tm_hour;
172         Serial.print("Check Save Time: ");
173         Serial.println(preHour);
174     }
175     // Do Save Image to Google Drive
176     if ( doFlag ) {
177         // ##### Get JPEG picture #####
178         Serial.println("Start get JPG");
179         getCameraJPEG();
180         // ##### get Access Token #####
181         Serial.println("Start get AccessToken");
182         getAccessToken();
183         // ##### Save JPEG to GoogleDrive #####
184         Serial.println("Start Post GoogleDrive");
185         postGoogleDriveByAPI();
186     }
187     delay(1);
188 }
189
```

現在時間の取得

実行フラグ

分単位で経過時間をチェックするため、
0分を超えると60をプラス

定期起動
チェック

設定値がプラスで、かつ、前回から
経過時間を過ぎていた場合に実行。
(もしくは前回時間が-1の場合)

前回時間から経過(異なり)していて、
設定値と同じ、かつ、分がゼロの場合
に実行

定刻起動
チェック

画像取得及びGoogleドライブ保存の
処理実行

8. 動作確認

GoogleのFolder-ID (プログラムで設定)

Name	Owner	Last modified	File size
M5TimerCam_20230210_150015.jpg	me	3:00 PM me	32 KB
M5TimerCam_20230210_152002.jpg	me	3:20 PM me	30 KB
M5TimerCam_20230210_154002.jpg	me	3:40 PM me	31 KB
M5TimerCam_20230210_160002.jpg	me	4:00 PM me	31 KB
M5TimerCam_20230210_162002.jpg	me	4:20 PM me	32 KB
M5TimerCam_20230210_164003.jpg	me	4:40 PM me	31 KB
M5TimerCam_20230210_170002.jpg	me	5:00 PM me	29 KB
M5TimerCam_20230210_172002.jpg	me	5:20 PM me	31 KB
M5TimerCam_20230210_174002.jpg	me	5:40 PM me	20 KB
M5TimerCam_20230210_180003.jpg	me	6:00 PM me	29 KB

定刻の15時及び20分毎に保存