Adding Timestamps from the SparkFun Gps

There is a lot of data that we ignored on page 45. Let's implement the time data. We will modify the code from pages 46 & 47

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the section iTOW timestamps in Integration manual for details.
4	U2	-	year *	y	Year (UTC)
6	U1	-	month *	month	Month, range 1..12 (UTC)
7	U1	-	day *	d	Day of month, range 1..31 (UTC)
8	U1	-	hour *	h	Hour of day, range 0..23 (UTC)
9	U1	-	min *	min	Minute of hour, range 0..59 (UTC)
10	U1	-	sec	s	Seconds of minute, range 0..60 (UTC)

```
typedef struct
{
  uint16_t year;
  uint8_t month;
  uint8_t day;
  uint8_t hour;
  uint8_t minute;
  uint8_t second;
  long lat;
  long lon;
  uint8_t SIV;
  long groundSpeed;
  long heading;
} NavData;
```

Here, we modify the struct to include the data we want. Remember that a struct is just a way of grouping like variables together under a common name.

```
----------
6/17/2024 19:55:20
Heading: 0.00
Lat: 334678920
Lon: -819904501
SIV: 4
gSp: 197
Hea: 6601319
----------
```

Results after neatly formating.

```
if (checksumA == CK_A && checksumB == CK_B)
{
  NavData data;
  data.year = payload[5]<<8 | payload[4];
  data.month = payload[6];
  data.day = payload[7];
  data.hour = payload[8];
  data.minute = payload[9];
  data.second = payload[10];
  data.lat = payload[31] <<24 | payload[30] << 16 | payload[29]<<8 | payload[28];
  data.lon = payload[27] <<24 | payload[26] << 16 | payload[25]<<8 | payload[24];
  data.SIV = payload[23];
  data.groundSpeed = payload[63] << 24 | payload[62] << 16 | payload[61] << 8 | payload[60];
  data.heading = payload[67] << 24 | payload[66] << 16 | payload[65] << 8 | payload[64];
  return data;
}
```

grab the data from the payload.