EPICS
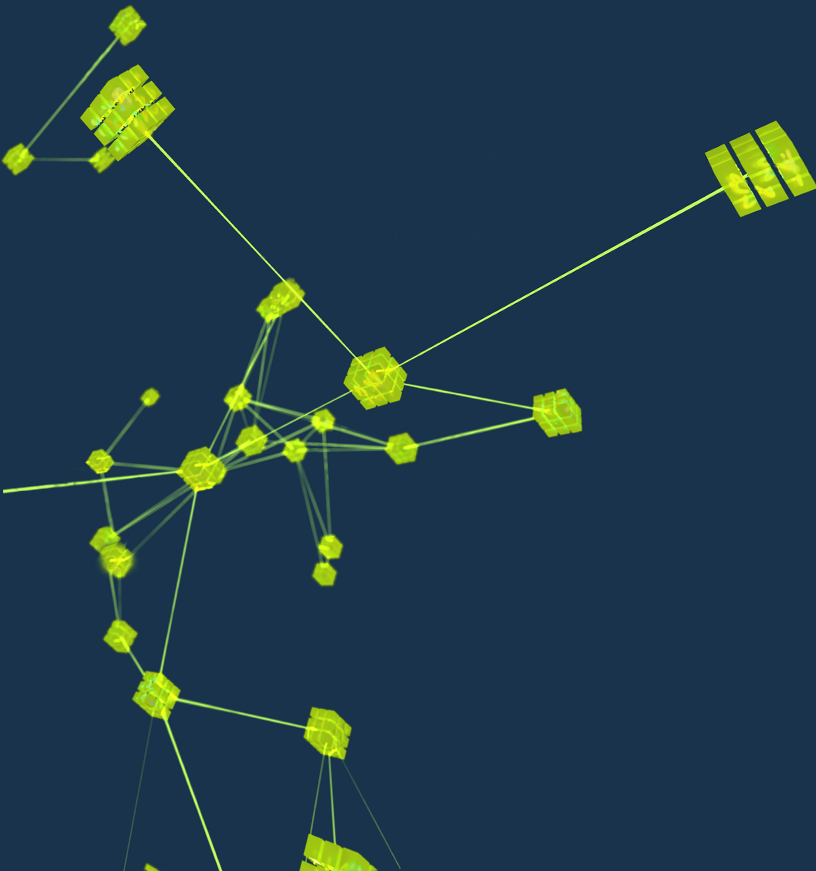
A electronic balance for beehives
(a more private project)

- A hive scale is an important tool which gives assessment if food consumption has been high and whether there is a need for feeding

- It is important to know how long the winter storage is, in addition it gives a very good indication of periods without any nectar flow

The central module is placed between to wooden plates under the hive and consists of a Arduino WeMos D1 (including WiFi), a temperature sensor, a singe point load cell for weighting purposes (BOSCHE H30A), a load cell amplifier (HX711), and a pb accu (4V)
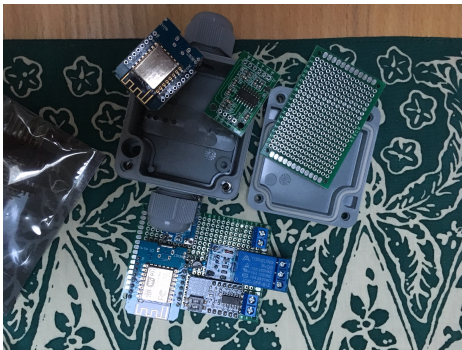
- With the deep sleep function of the Arduino platform one can maximize the battery life. In deep sleep it consumes just 40 µA.

- WeMos D1 is a ESP8266 WiFi based board

- It's super easy to program


```
void loop() {

double Weight;
double Temp;
char buffer[64];
  scale.power_up();
  Weight = scale.get_units(20);
  scale.power_down();
  Weight -= bbbOffset;
  Temp = readTemp(tAddr, ds);
  /* Gruen (4711), gelb (4712), blau (4713), braun (4714) gelb, braun, */
  sprintf(buffer,"%f,%f\n", Weight, Temp);
  int res;
  UDP.beginPacket(bienenwaagenIp, bienenwaagenPort);
  UDP.write(buffer);
  UDP.endPacket();
  yield();
  delay (1000);
  ESP.deepSleep(sleepSeconds * 1000000, WAKE_RF_DEFAULT );
}
```

- Arduino sends UDP messages to an IOC running on a VServer

- IOC receives data and "put" them into PV's

```
epics@bienenwaagen:~/bb/BBB/bbbApp/Db$ cat  bbb.db
record(waveform,"$(P)"){
  field(DTYP,"UDP Intr")
  field(DESC,"$(D)")
  field(SCAN,"I/O Intr")
  field(INP,"$(PORT)")
  field(FTVL, "DOUBLE")
  field(NELM, "4")
}

record(subArray, "$(P)[0]") {
  field(INP, "$(P) CP")
  field(INDX, "0")
  field(MALM, "4")
  field(NELM, "1")
  field(FTVL, "DOUBLE")
  field(FLNK, "$(P):Weight")
}

record(ai, "$(P):Weight") {
  field(INP, "$(P)[0].VAL")
  field(EGU,"Kg")
}

record(subArray, "$(P)[1]") {
  field(INP, "$(P) CP")
  field(INDX, "1")
  field(MALM, "4")
  field(NELM, "1")
  field(FTVL, "DOUBLE")
  field(FLNK, "$(P):Temp")
}

record(ai, "$(P):Temp") {
  field(INP, "$(P)[1].VAL")
  field(EGU,"gradC")
}
```

```
epicsFloat64 *dbuf;

  while(1) {

    if(!buf) {
      /* allocate and initialize a new buffer for later (local) use */
      buf = callocMustSucceed(1, nbytes, "buffer");
      dbuf = (epicsFloat64*)buf;
    }

    // try to receive some data, this is a blocking call
    if ((recv_len = recvfrom(s, receiveBuf, RECV_BUFLEN, 0,
              (struct sockaddr *) &si_other, &slen)) == -1)
    {
        perror("recvfrom()");
        return;
    }
    receiveBuf[recv_len] = 0;
    /* print details of the client/peer and the data received */
      printf("Received packet from %s:%d\n", inet_ntoa(si_other.sin_addr),
              ntohs(si_other.sin_port));
printf("received: %s\n", receiveBuf);
    if (nbytes >= 2 * sizeof(double)) {
      sscanf (receiveBuf, "%lf,%lf", &dbuf[0], &dbuf[1]);
      printf("scanf : %lf, %lf\n", dbuf[0], dbuf[1]);
    }

    epicsMutexMustLock(priv->lock);
    if(!priv->nextBuffer) {
      /* make the local buffer available to the read_wf function */
      priv->nextBuffer = buf;
      buf = NULL;
      priv->numBytes = priv->maxBytes;
    }
    epicsMutexUnlock(priv->lock);

    scanIoImmediate(priv->scan, priorityHigh);
```

```
epics@bienenwaagen:~/bb/BBB$ ps -ef | grep proc
bbb      1379     1  0 Jun05 ?       00:00:50 /usr/bin/procServ -q -c /epics/iocs/bbb
 -i ^D^C^] -p /var/run/softioc-bbb.pid -n bbb --restrict --logfile=/var/log/softioc/bbb
/bbb.log --coresize= 4051 /epics/iocs/bbb/st.cmd
```

- To archive the incoming data archiver appliance is used

- To visualize date on iPhone/iPad Daviz is used
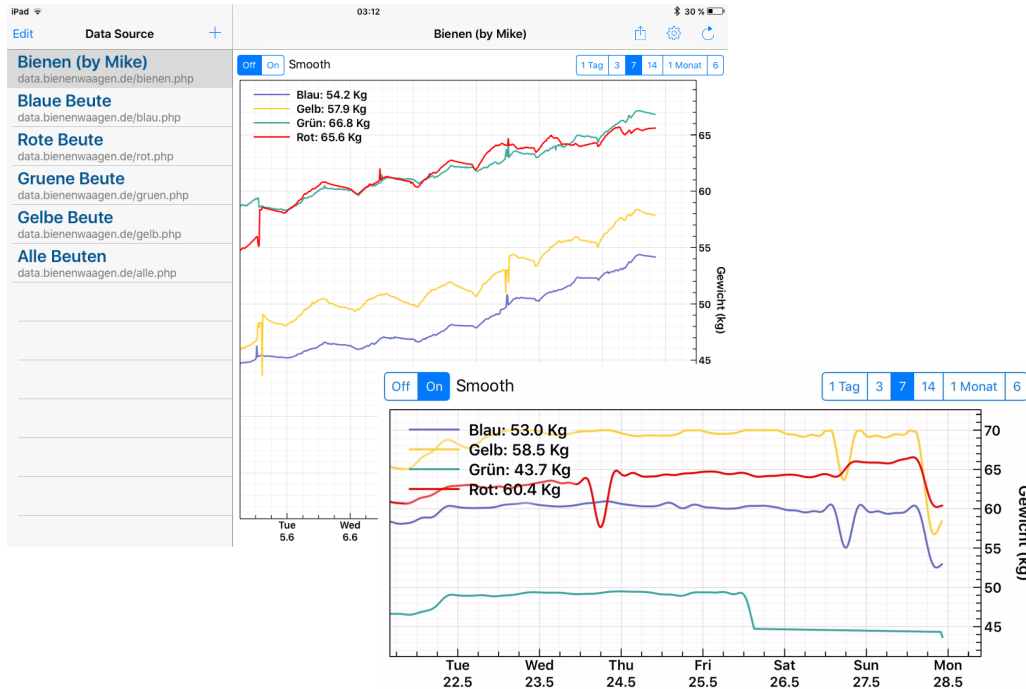
- data read from the archiver on demand by JSON like

  $url = "http://185.228.137.144:17668/retrieval/data/getData.json?

  pv=median_1800(BBB%3AGelb%3ATemp)&from=$old&to=$now";

  $json = file_get_contents($url);