# intoft25_2023_intoft25_p4_01 / Prototype6_Data_Meetkast_naar_CU / Code / ESP32 / ESP32.ino @ 312

View History Annotate

```
#include <SPI.h>
#include <mcp2515.h>
#include <stdint.h>

const byte DIFFERENTPATIENTAMOUNT = 2;
const byte DIFFERENTMACHINEAMOUNT = 7;
long patientData[DIFFERENTPATIENTAMOUNT][DIFFERENTMACHINEAMOUNT];
const uint8_t ROWS = 14;
const uint8_t AMOUNTOFCONFIGVALUES = 4;
const uint8_t TEMPPATIENTID = 0;

enum dataValues {
  ECG = 0,
  HEMOGLOBIN = 1,
  CHOLESTEROL =2,
  UPPERPRESSURE = 3,
  NEGATIVEPRESSURE = 4,
  OXYGENLEVEL = 5,
  HEARTBEAT = 6
};


enum dataType {
  CANID,
  FREQUENCY,
  RESOLUTION,
  PORT
};

struct can_frame canMsg;
```

```
MCP2515 mcp2515(2); // Has to be something outside 6-11


void setup() {
  Serial.begin(115200);

  mcp2515.reset();
  mcp2515.setBitrate(CAN_125KBPS, MCP_8MHZ);
  mcp2515.setNormalMode();

  Serial.println("------- CAN Read ----------");
  Serial.println("ID  DLC   DATA");
}

void loop() {
  /*Voorbeeld van hoe dit geprint wordt:
  ID: 4 CharAmount: 8   -  Data 0: 80  -  Data 1: 0  -  Data 2: 0  -  Data 3: 0  -  Data 4: 0  -  Data 5: 0  -  Data 6: 0
  Hierin is ID het ID van het CAN bericht én ook het machineID.
  CharAmount is CAN_DLC, dus hoeveel data chars/ bits er zijn.
  Voor alle data is het op de tijd van schrijven (06-05-2024) zo dat het nummer de patiëntID is. Dus in het voorbeeldje h
  */
  if (mcp2515.readMessage(&canMsg) == MCP2515::ERROR_OK) {
     Serial.print("ID: ");
     Serial.print(canMsg.can_id, HEX); // print CAN_ID
     Serial.print(" ");
     Serial.print("CharAmount: ");
     Serial.print(canMsg.can_dlc, HEX); // print CAN_DLC (amount of data characters/bits)
     Serial.print(" ");

    for (int i = 0; i < canMsg.can_dlc; i++)  {  // print all CAN_msg.data[] values
   // Serial.println(canMsg.data[i],HEX);
      Serial.print("  -  Data ");
      Serial.print(i);
      Serial.print(": ");
      Serial.print(canMsg.data[i],DEC); // Will print 208 when overflow happens
    }
    Serial.println();
  }

  for(uint32_t row = 0 ; row < ROWS ; row++)
  {
    if(row == CANID){
      for(uint32_t column = 0 ; column < AMOUNTOFCONFIGVALUES; column++)
      {
        switch(column)
          {
            case ECG:
              patientData[TEMPPATIENTID][0] = column;
              break;
            case HEMOGLOBIN:
              patientData[TEMPPATIENTID][1] = column;
              break;
            case CHOLESTEROL:
              patientData[TEMPPATIENTID][2] = column;
              break;
            case UPPERPRESSURE:
              patientData[TEMPPATIENTID][3] = column;
              break;
            case NEGATIVEPRESSURE:
              patientData[TEMPPATIENTID][4] = column;
              break;
            case OXYGENLEVEL:
              patientData[TEMPPATIENTID][5] = column;
              break;
            case HEARTBEAT:
              patientData[TEMPPATIENTID][6] = column;
              break;
          }
      }
    }
  }

//  for (int patientId = 0; patientId < DIFFERENTPATIENTAMOUNT; ++patientId) {
    for (int machineId = 0; machineId < DIFFERENTMACHINEAMOUNT; ++machineId) {
      Serial.print(patientData[TEMPPATIENTID][machineId]);
      Serial.print(" ");
    }
  }
  Serial.println();
//}
```