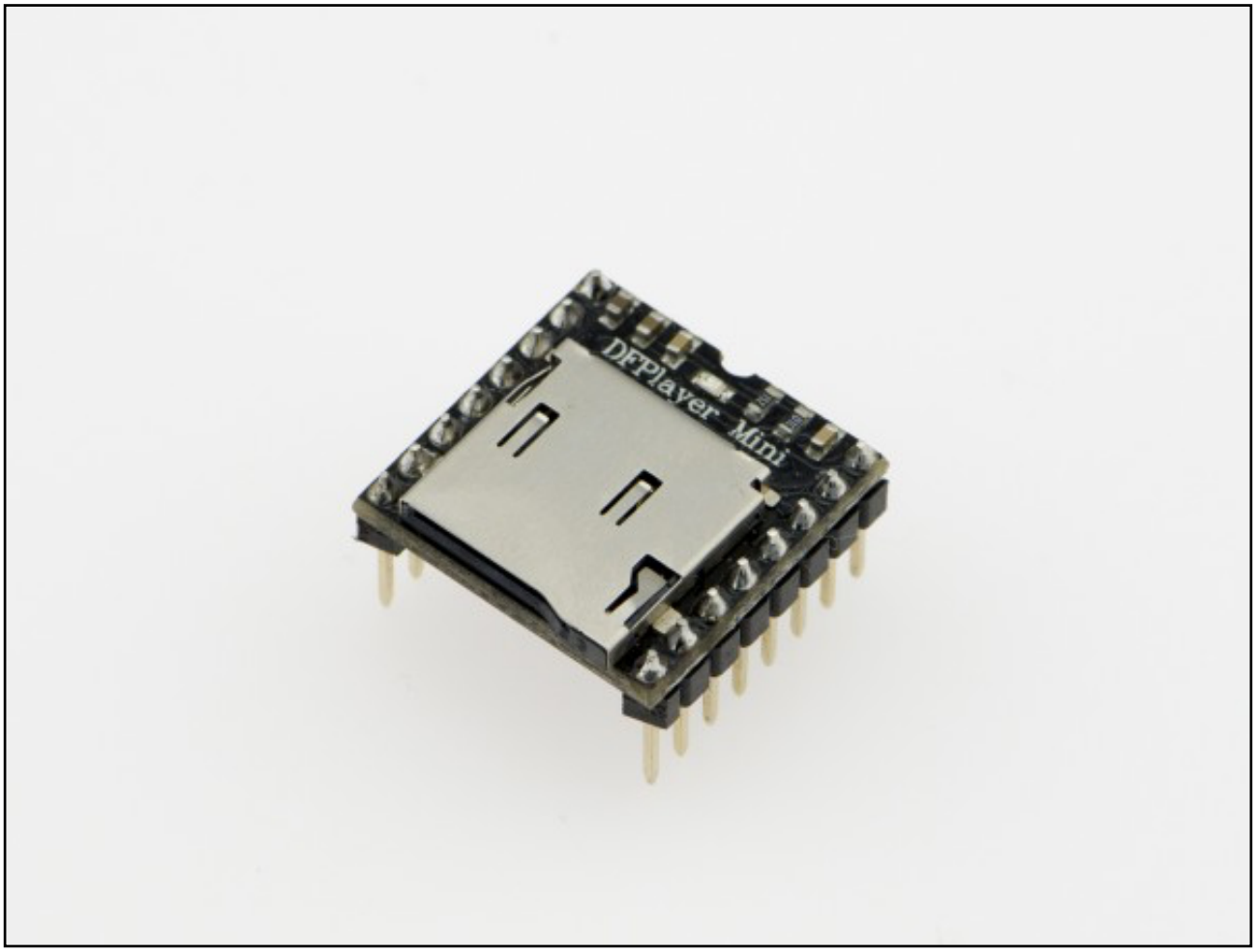
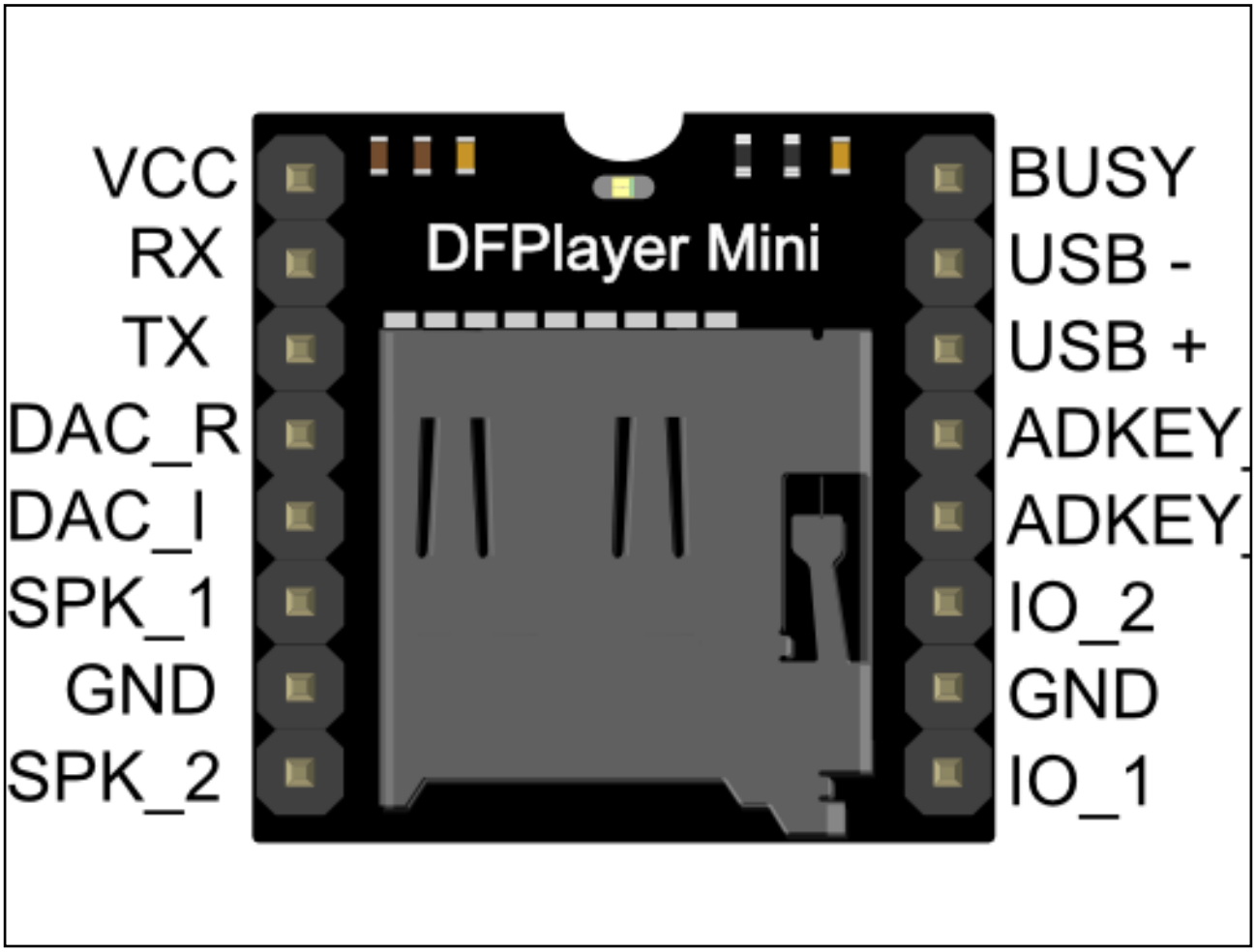
# Basteltruppe 8

# Your Projects

Jasmin
Katzenbuzzer


DFPlayer Mini


VCC
RX
TX
DAC_R
DAC_I
SPK_1
GND
SPK_2
DFPlayer Mini
BUSY
USB -
USB +
ADKEY
ADKEY
IO_2
GND
IO_1

# To Buy

- Speaker
- Arduino Nano (?)
- Power supply: Postponed
- Button: 3D Print + STSP Switch

ARDUINO
WORD CLOCK
Christoph

# To Buy

- LED Strip
- Power supply für LED strip
- Real Time Clock für Arduino

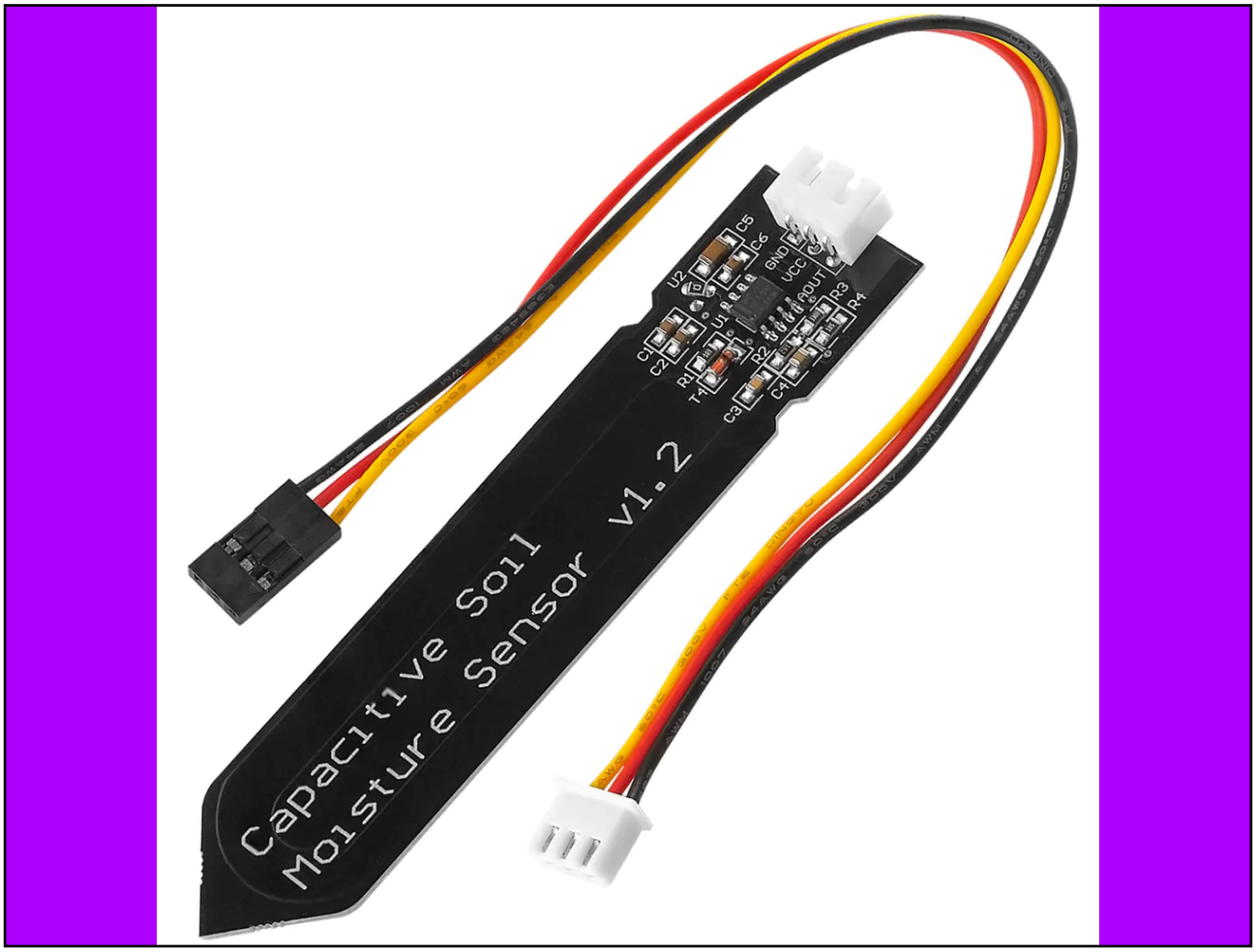
Saadet
Katzenalarmanlage

# To Buy

- Ultrasonic Sensor
- Beeper (Cat frequency)
- LCD Display
- Distance buttons
- Power supply: Postponed

Silas
Arduino
Gardener

FCD-180B
AC220V/240V
50/60hz
27±5mA CLASSE


Capacitive Soil
Moisture Sensor V1.2
GND
VCC
AOUT
U2
U1
C1
C2
C3
C4
C5
C6
R1
R2
R3
R4
T1

# To Buy

Solenoid Valve
Bodenfeuchtesensor
5V Relay
Batterie für Relay
Power Supply: Erstmal USB

# Karsten
# Lichtwecker mit RGB Stripes

# To buy

- LCD Display
- LED Stripes
- Real time clock
- Power supply: Postponed

# Stefan

- Spracherkennung, fragen bei ES
- Soll Kommando erkennen

# Melik

- Zauberstab
- TODO Peter: Wie?
- Soundstation wie bei Jasmins Projekt
- Power supply: Postponed

# Lars
# Wetterstation

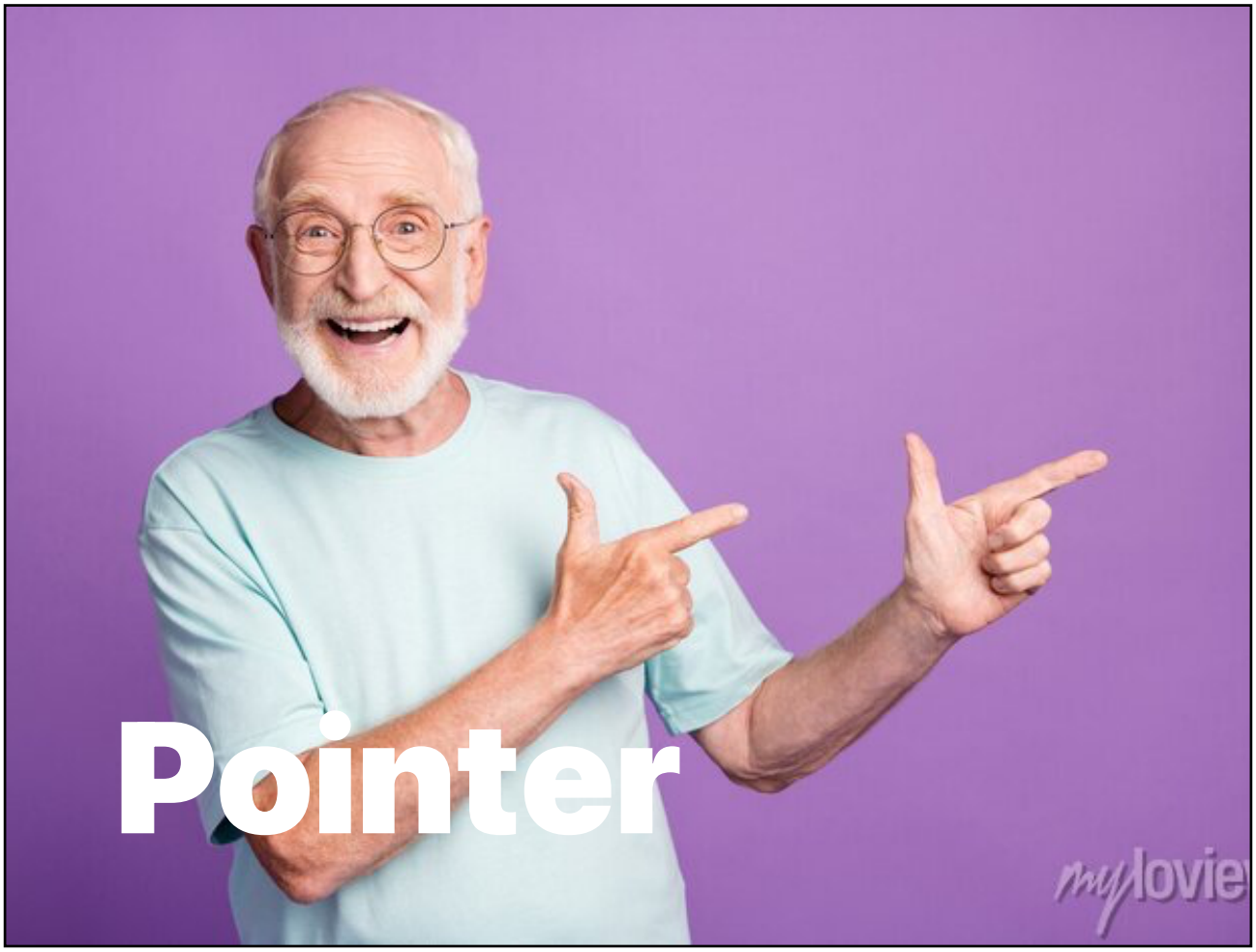
# To buy

- LCD Display
- Arduino Nano
- BME 280

# Fabio

- Ferngesteuertes Auto
- Maybe an mini Flitzern orientieren? TODO!

Content!


Pointer

Sometimes we don't want to copy memory around

Your memory exists and is here: do something

# How to differentiate between a value and an address?

# Types!

```
int main(){
    int x = 5;
    uint8_t y = 6;
    float pff = 12.5f;
    int* x_ptr = &x;
    uint8_t* y_pointer = &y;
    float* pff_pointah = &pff;
}
```

# & means "the address of"

```
void x(){
    int* x_ptr = &x;
    uint8_t* y_pointer = &y;
    float* pff_pointah = &pff;
}
```

* means "this can be dereferenced and interpreted as"

# Dereferencing

```
void setup(){
    Serial.begin(9600);
    int x = 5;
    int* y = &x;
    *y = 6;
    Serial.println(x);
    Serial.println(*y);
}
```

# interpreted as

```
void setup(){
    float x = 5.5f;
    uint32_t* y =(uint32_t* ) &x;
    *y = 6;
    Serial.println(x);
    Serial.println(*y);
}
```

# Beware of fooling around

```
void setup(){
    int* x;
    *x = 5; // ERROR
    x = 4; // Oh oh
    int y;
    x = &y;
    if ( x < 5){
        //oh oh
    }
}
```