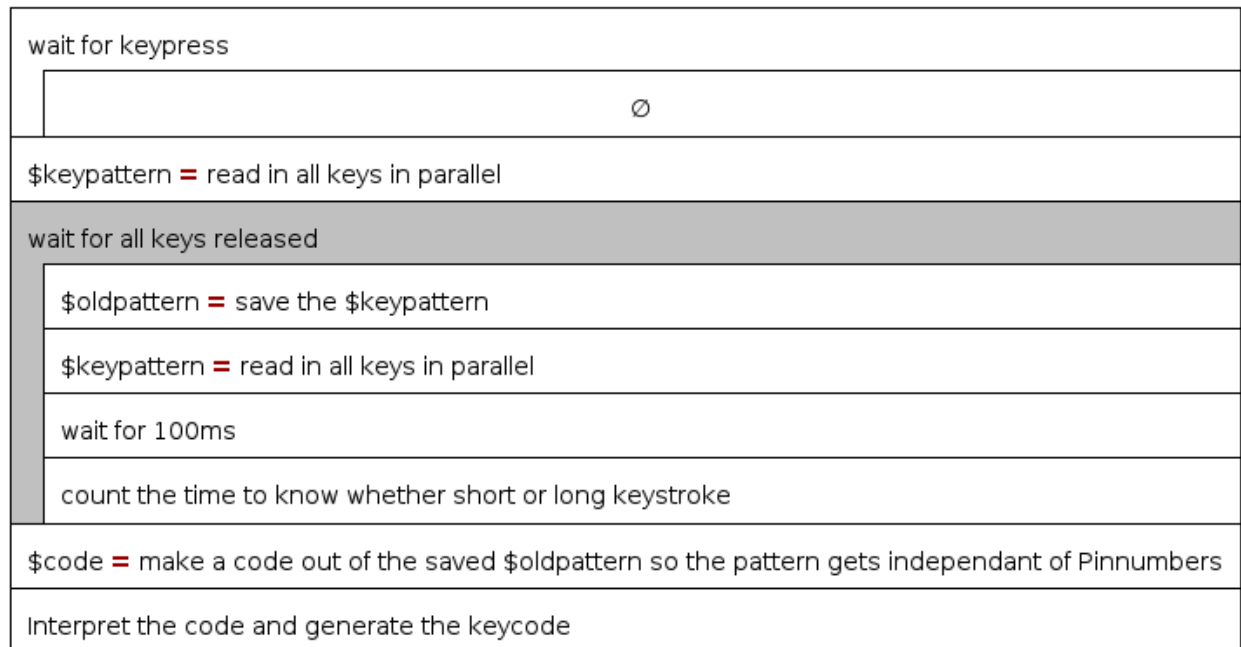


# Keyboard

Read in the keyboard

## Keyboard

Get keycode for short/long single/multiple key pressures



== main.cpp ==

```
#include <Arduino.h>
#include "keyboard.h"
#undef DEBUG

keycodes_t key1 = keycode_invalide;

void setup()
{
    activatePullups();
    Serial.begin(9600);
}

void loop()
{
    key1 = readKeys(); //blockiert!
    uint8_t x=(int)key1;
    Serial.print(x,HEX);
}
```

== keyboard.h ==

```
#ifndef KEYBOARD_H_
#define KEYBOARD_H_

//Keys on PORTX/PINX
#define PINX PIND
#define PORTX PORTD
#define VOL_DN 0
#define VOL_UP 1
```

```

#define CH_DN 4
#define CH_UP 7

#ifdef DEBUG
#define DEBOUNCETIME 2
#define LONGKEYPRESSURE_MINIMALTIME 5
#else
#define DEBOUNCETIME 25
#define LONGKEYPRESSURE_MINIMALTIME 15
#endif

enum keycodes_t{
    keycode_invalide,
    keycode_vol_up, keycode_vol_dn,
    keycode_ch_up,    keycode_ch_dn,
    keycode_vol_up_long, keycode_vol_dn_long,
    keycode_ch_up_long,    keycode_ch_dn_long, keycode_mute,
    keycode_ch_search, keycode_ch_save, keycode_eeprom_clear, keycode_testmode
};

keycodes_t readKeys();
void activatePullups();

#endif /* KEYBOARD_H_ */
== keyboard.cpp ==

#include <Arduino.h>
#include "keyboard.h"

void activatePullups(){
    PORTX |= (1<<VOL_UP|1<<VOL_DN|1<<CH_UP|1<<CH_DN);
}

enum keycodes_t readKeys(){
    uint8_t mask = 1<<VOL_UP|1<<VOL_DN|1<<CH_UP|1<<CH_DN;
    while ((PINX & mask) == mask){} //wait for key pressure
    delay(DEBOUNCETIME); //debounce time
    ...
}

```

## **Analog Keyboard**

Jede Taste erzeugt an einem Spannungsteiler eine eigene Spannung. Über einen Analogeingang werden die Spannungen ausgewertet.  
 Mehrfachastendrucke möglich (aber nicht getestet)

```

//short Keystrokes = Keycode 1,2,3
//long Keystrokes = 11, 12, 13
//very long Keystrokes = 101, 102, 103

uint8_t waitForKey() {
    int keyTime = 0;
    TRACE ("waitForKey")
    int newSensorValue = 0;
    DELAY (20); //let the adc recreate

    while (sensorValue == 0) {
        sensorValue = analogRead (analogInPin);
    }
}

```

```

    DELAY (50) //debounce
}

//delay(50); //let the adc settle
sensorValue = analogRead (analogInPin);
DELAY (300);
newSensorValue = analogRead (analogInPin);

while ( (newSensorValue / 100 == sensorValue / 100) && (sensorValue != 0) )
{ // div by 100 to reduce sensibility
    DELAY (300)
    keyTime++;
    newSensorValue = analogRead (analogInPin);
    if (keyTime >= 2){
        Radio.mute(); //feedback to user for long key pressure
        delay(50);
        Radio.unMute();
    }
    if (keyTime > 6){
        Radio.mute(); //feedback to user for very long key pressure
        delay(500);
        Radio.unMute();
    }
}

int keyValue = map (sensorValue, 0, 1023, 0, 5);

if (keyTime >= 2 && keyTime <= 6)
{ keyValue += 10;
}
else if (keyTime > 6)
{ keyValue += 100;
}

sensorValue = 0;

return keyValue;
}

```