

Sender (10 Tasten)

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(2, 13); // pin2 an HC-05 TX, 13 an RX
const int pin12 = 12;
int pin12state = 0;
int lastpin12state = 0;

const int pin11 = 11;
int pin11state = 0;
int lastpin11state = 0;

const int pin10 = 10;
int pin10state = 0;
int lastpin10state = 0;

const int pin9 = 9;
int pin9state = 0;
int lastpin9state = 0;

const int pin8 = 8;
int pin8state = 0;
int lastpin8state = 0;

const int pin7 = 7;
int pin7state = 0;
int lastpin7state = 0;

const int pin6 = 6;
int pin6state = 0;
int lastpin6state = 0;

const int pin5 = 5;
int pin5state = 0;
int lastpin5state = 0;

const int pin4 = 4;
int pin4state = 0;
int lastpin4state = 0;

const int pin3 = 3;
int pin3state = 0;
int lastpin3state = 0;

void setup() {
  BTSerial.begin(9600);
  pinMode(pin12, INPUT_PULLUP);
  pinMode(pin11, INPUT_PULLUP);
  pinMode(pin10, INPUT_PULLUP);
  pinMode(pin9, INPUT_PULLUP);
  pinMode(pin8, INPUT_PULLUP);
  pinMode(pin7, INPUT_PULLUP);
  pinMode(pin6, INPUT_PULLUP);
  pinMode(pin5, INPUT_PULLUP);
  pinMode(pin4, INPUT_PULLUP);
  pinMode(pin3, INPUT_PULLUP);
}

void loop() {
  pin12state = digitalRead(pin12);
  pin11state = digitalRead(pin11);
  pin10state = digitalRead(pin10);
  pin9state = digitalRead(pin9);
  pin8state = digitalRead(pin8);
  pin7state = digitalRead(pin7);
  pin6state = digitalRead(pin6);
  pin5state = digitalRead(pin5);
  pin4state = digitalRead(pin4);
  pin3state = digitalRead(pin3);
  //Taste 1
  if (pin12state != lastpin12state) {
    if (pin12state == LOW) { //LOW, da Pull-up
      BTSerial.write('1');
    } else {
      BTSerial.write('0');
    }
    lastpin12state = pin12state;
    delay(50); // Entprellen
  }
  //Taste 2
  if (pin11state != lastpin11state) {
    if (pin11state == LOW) {
      BTSerial.write('2');
    } else {
      BTSerial.write('0');
    }
    lastpin11state = pin11state;
    delay(50); // Entprellen
  }
  //Taste 3
  if (pin10state != lastpin10state) {
    if (pin10state == LOW) { //LOW, da Pull-up
      BTSerial.write('3');
    } else {
      BTSerial.write('0');
    }
    lastpin10state = pin10state;
    delay(50); // Entprellen
  }
  //Taste 4
  if (pin9state != lastpin9state) {
    if (pin9state == LOW) { //LOW, da Pull-up
      BTSerial.write('4');
    } else {
      BTSerial.write('0');
    }
    lastpin9state = pin9state;
    delay(50); // Entprellen
  }
  //Taste 5
  if (pin8state != lastpin8state) {
    if (pin8state == LOW) { //LOW, da Pull-up
      BTSerial.write('5');
    } else {
      BTSerial.write('0');
    }
    lastpin8state = pin8state;
    delay(50); // Entprellen
  }
  //Taste 6
  if (pin7state != lastpin7state) {
    if (pin7state == LOW) { //LOW, da Pull-up
      BTSerial.write('6');
    } else {
      BTSerial.write('0');
    }
    lastpin7state = pin7state;
    delay(50); // Entprellen
  }
  //Taste 7
  if (pin6state != lastpin6state) {
    if (pin6state == LOW) { //LOW, da Pull-up
      BTSerial.write('7');
    } else {
      BTSerial.write('0');
    }
    lastpin6state = pin6state;
    delay(50); // Entprellen
  }
  //Taste 8
  if (pin5state != lastpin5state) {
    if (pin5state == LOW) { //LOW, da Pull-up
      BTSerial.write('8');
    } else {
      BTSerial.write('0');
    }
    lastpin5state = pin5state;
    delay(50); // Entprellen
  }
  //Taste 9
  if (pin4state != lastpin4state) {
    if (pin4state == LOW) { //LOW, da Pull-up
      BTSerial.write('A');
    } else {
      BTSerial.write('0');
    }
    lastpin4state = pin4state;
    delay(50); // Entprellen
  }
  //Taste 10
  if (pin3state != lastpin3state) {
    if (pin3state == LOW) { //LOW, da Pull-up
      BTSerial.write('B');
    } else {
      BTSerial.write('0');
    }
    lastpin3state = pin3state;
    delay(50); // Entprellen
  }
}
```

Empfänger (5 Motoren)

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(2, 13); // Pin2 an HC-05 TX, 13 an RX
const int mot1pin12 = 12;
const int mot1pin11 = 11;
const int mot1pin10 = 10;
const int mot1pin9 = 9;
const int mot2pin8 = 8;
const int mot2pin7 = 7;
const int mot2pin6 = 6;
const int mot2pin5 = 5;
const int mot3pin4 = 4;
const int mot3pin3 = 3;

void setup() {
  BTSerial.begin(9600);
}

void loop() {
  if (BTSerial.available()) {
    char command = (char)BTSerial.read();
    //-----Motor 1-----
    if (command == '1') {
      analogWrite(mot1pin12, 150);
      analogWrite(mot1pin11, 0);
    } else {
      analogWrite(mot1pin12, 0);
    }
    if (command == '2') {
      analogWrite(mot1pin12, 0);
      analogWrite(mot1pin11, 150);
    } else {
      analogWrite(mot1pin11, 0);
    }
    //-----Motor 2-----
    if (command == '3') {
      analogWrite(mot1pin10, 0);
      analogWrite(mot1pin9, 150);
    } else {
      analogWrite(mot1pin9, 0);
    }
    if (command == '4') {
      analogWrite(mot1pin10, 150);
      analogWrite(mot1pin9, 0);
    } else {
      analogWrite(mot1pin10, 0);
    }
    //-----Motor 3-----
    if (command == '5') {
      analogWrite(mot2pin8, 150);
      analogWrite(mot2pin7, 0);
    } else {
      analogWrite(mot2pin8, 0);
    }
    if (command == '6') {
      analogWrite(mot2pin8, 0);
      analogWrite(mot2pin7, 150);
    } else {
      analogWrite(mot2pin7, 0);
    }
    //-----Motor 4-----
    if (command == '7') {
      analogWrite(mot2pin6, 150);
      analogWrite(mot2pin5, 0);
    } else {
      analogWrite(mot2pin6, 0);
    }
    if (command == '8') {
      analogWrite(mot2pin6, 0);
      analogWrite(mot2pin5, 150);
    } else {
      analogWrite(mot2pin5, 0);
    }
    //-----Motor 5-----
    if (command == 'A') {
      analogWrite(mot3pin4, 150);
      analogWrite(mot3pin3, 0);
    } else {
      analogWrite(mot3pin4, 0);
    }
    if (command == 'B') {
      analogWrite(mot3pin4, 0);
      analogWrite(mot3pin3, 150);
    } else {
      analogWrite(mot3pin3, 0);
    }
    if (command == '0') {
      analogWrite(mot1pin12, 0);
      analogWrite(mot1pin11, 0);
      analogWrite(mot1pin10, 0);
      analogWrite(mot1pin9, 0);
      analogWrite(mot2pin8, 0);
      analogWrite(mot2pin7, 0);
      analogWrite(mot2pin6, 0);
      analogWrite(mot2pin5, 0);
      analogWrite(mot3pin4, 0);
      analogWrite(mot3pin3, 0);
    }
  }
}
```