

Tuerschloss NFC

L. Mazzoleni

23. Juni 2016

Inhaltsverzeichnis

1 Code

2

1 Code

```

1 //tuerschloss Luca Mazzoleni 09.07.2015
2 //Arduino Leonardo
3
4 //Erweiterung auf NFC-TAGs 22.06.2016
5
6 //library https://github.com/adafruit/Adafruit-PN532
7
8 #include <Wire.h>
9 #include <SPI.h>
10 #include <Adafruit_PN532.h>
11
12 #define button0 (2)
13 #define button1 (10)
14 #define button2 (0)
15 #define button3 (11)
16 #define button4 (7)
17 #define button5 (8)
18 #define button6 (9)
19 #define button7 (4)
20 #define button8 (5)
21 #define button9 (6)
22 #define buttonStar (1)
23 #define buttonHash (3)
24 #define ledOpen (12)
25 #define ledClose (13)
26 #define doorPin (A0)
27 //NFC
28 #define PN532_SCK (A1)
29 #define PN532_MOSI (A2)
30 #define PN532_SS (A3)
31 #define PN532_MISO (A4)
32
33 #define PN532_IRQ (A1)
34 #define PN532_RESET (A2) // Not connected by default on the NFC Shield
35
36 //SPI connection
37 Adafruit_PN532 nfc(PN532_SCK,
38                    PN532_MISO,
39                    PN532_MOSI,
40                    PN532_SS);
41
42 //Bei PW wechsel k und secretCode anpassen !! oder mit Sizeof Array lösen
43 const int k = 4; //Passwordlength
44 const int maxIN = (10 + 1); //Max Input
45 char secretCode[k] = {'1', '1', '1', '1'}; //Code anpassen
46 char inputCode[maxIN];
47
48 #if defined(ARDUINO_ARCH_SAMD)
49 // for Zero, output on USB Serial console, remove line below if using programming port to program
   the Zero!
50 // also change #define in Adafruit_PN532.cpp library file
51 #define Serial SerialUSB
52 #endif
53
54 void setup()
55 {
56     Serial.begin(115200);
57
58     pinMode(ledOpen, OUTPUT);
59     pinMode(ledClose, OUTPUT);
60     pinMode(doorPin, INPUT);
61     pinMode(button0, INPUT);
62     pinMode(button1, INPUT);

```

```

63 pinMode(button2, INPUT);
64 pinMode(button3, INPUT);
65 pinMode(button4, INPUT);
66 pinMode(button5, INPUT);
67 pinMode(button6, INPUT);
68 pinMode(button7, INPUT);
69 pinMode(button8, INPUT);
70 pinMode(button9, INPUT);
71 pinMode(buttonStar, INPUT);
72 pinMode(buttonHash, INPUT);
73 digitalWrite(ledOpen, LOW);
74 digitalWrite(ledClose, LOW);
75 analogWrite(doorPin, 0);
76 digitalWrite(button0, HIGH);
77 digitalWrite(button1, HIGH);
78 digitalWrite(button2, HIGH);
79 digitalWrite(button3, HIGH);
80 digitalWrite(button4, HIGH);
81 digitalWrite(button5, HIGH);
82 digitalWrite(button6, HIGH);
83 digitalWrite(button7, HIGH);
84 digitalWrite(button8, HIGH);
85 digitalWrite(button9, HIGH);
86 digitalWrite(buttonStar, HIGH);
87 digitalWrite(buttonHash, HIGH);

88 Serial.println("Hello!");

89 nfc.begin();

90
91 uint32_t versiondata = nfc.getFirmwareVersion();
92 if (!versiondata)
93 {
94     Serial.print("Didn't find PN53x board");
95     // while (1)
96     //     ; // stop if Board not found
97 }
98 // Got ok data, print it out!
99 Serial.print("Found chip PN5");
100 Serial.println((versiondata >> 24) & 0xFF, HEX);
101 Serial.print("Firmware ver. ");
102 Serial.println((versiondata >> 16) & 0xFF, DEC);
103 Serial.print('.');
104 Serial.println((versiondata >> 8) & 0xFF, DEC);

105 // configure board to read RFID tags
106 nfc.SAMConfig();

107
108 Serial.println("Waiting for an ISO14443A Card ...");
109 }

110
111 void loop()
112 {
113     int n; //VarCountNumber
114     int i; //VarInputCode
115     int correct = 0; //VarCountPW
116     int p = 0; //VarCountlengthIN

117     uint8_t success;
118     uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0}; // Buffer to store the returned UID
119     uint8_t uidLength; // Length of the UID (4 or 7 bytes depending on ISO14443A card type)

120
121     void accesgranted();
122     void accesdenied();
123     void reset(); //reset Inputcode-array

```

```

129 _Bool checkid(double idcard);
131 //
132 //=====
133 //ABFRAGE NFC
134 Serial.println("Abfrage NFC");
135 // Wait for an ISO14443A type cards (Mifare, etc.). When one is found
136 // 'uid' will be populated with the UID, and uidLength will indicate
137 // if the uid is 4 bytes (Mifare Classic) or 7 bytes (Mifare Ultralight)
138 success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid, &uidLength, 100); // 100 is the time
139 // for a Timeout
140
141 if (success)
142 {
143     // Display some basic information about the card
144     Serial.println("Found an ISO14443A card");
145     Serial.print("  UID Length: ");
146     Serial.print(uidLength, DEC);
147     Serial.println(" bytes");
148     Serial.print("  UID Value: ");
149     nfc.PrintHex(uid, uidLength);
150
151     if (uidLength == 4)
152     {
153         // We probably have a Mifare Classic card ...
154         uint32_t cardid = uid[0];
155         cardid <<= 8;
156         cardid |= uid[1];
157         cardid <<= 8;
158         cardid |= uid[2];
159         cardid <<= 8;
160         cardid |= uid[3];
161         Serial.print("Seems to be a Mifare Classic card #");
162         Serial.println(cardid);
163
164         if (checkid(cardid))
165         {
166             accesgranted();
167         }
168         else
169         {
170             accesdenied();
171         }
172     }
173     Serial.println("");
174 }
175 //
176 //=====
177 //ABFRAGE ZAHLENFELD
178 //TODO TIMEOUT
179 //TODO MATRIX
180 Serial.println("Abfrage Zahlenfeld");
181
182 if (digitalRead(buttonStar) == LOW)
183 {
184     reset();
185     while (1)
186     {
187         if (digitalRead(buttonStar) == HIGH)
188         {
189             Serial.println("--*--");
190             delay(1);
191         }
192     }
193 }

```

```
do
{
    if (digitalRead(button0) == LOW)
    {
        while (1)
        {
            if (digitalRead(button0) == HIGH)
            {
                inputCode[p] = ('0');
                Serial.println("--0--");
                delay(1);
                ++p;
                break;
            }
        }
    }

    if (digitalRead(button1) == LOW)
    {
        while (1)
        {
            if (digitalRead(button1) == HIGH)
            {
                inputCode[p] = ('1');
                Serial.println("--1--");
                delay(1);
                ++p;
                break;
            }
        }
    }

    if (digitalRead(button2) == LOW)
    {
        while (1)
        {
            if (digitalRead(button2) == HIGH)
            {
                inputCode[p] = ('2');
                Serial.println("--2--");
                delay(1);
                ++p;
                break;
            }
        }
    }

    if (digitalRead(button3) == LOW)
    {
        while (1)
        {
            if (digitalRead(button3) == HIGH)
            {
                inputCode[p] = ('3');
                Serial.println("--3--");
                delay(1);
                ++p;
                break;
            }
        }
    }

    if (digitalRead(button4) == LOW)
    {
        while (1)
        {
            if (digitalRead(button4) == HIGH)
            {
                inputCode[p] = ('4');
                Serial.println("--4--");
                delay(1);
```

```
253         ++p;  
254         break;  
255     }  
256 }  
257  
258 if (digitalRead(button5) == LOW)  
259 {  
260     while (1)  
261     {  
262         if (digitalRead(button5) == HIGH)  
263         {  
264             inputCode[p] = ('5');  
265             Serial.println("--5--");  
266             delay(1);  
267             ++p;  
268             break;  
269         }  
270     }  
271  
272 if (digitalRead(button6) == LOW)  
273 {  
274     while (1)  
275     {  
276         if (digitalRead(button6) == HIGH)  
277         {  
278             inputCode[p] = ('6');  
279             Serial.println("--6--");  
280             delay(1);  
281             ++p;  
282             break;  
283         }  
284     }  
285  
286 if (digitalRead(button7) == LOW)  
287 {  
288     while (1)  
289     {  
290         if (digitalRead(button7) == HIGH)  
291         {  
292             inputCode[p] = ('7');  
293             Serial.println("--7--");  
294             delay(1);  
295             ++p;  
296             break;  
297         }  
298     }  
299  
300 if (digitalRead(button8) == LOW)  
301 {  
302     while (1)  
303     {  
304         if (digitalRead(button8) == HIGH)  
305         {  
306             inputCode[p] = ('8');  
307             Serial.println("--8--");  
308             delay(1);  
309             ++p;  
310             break;  
311         }  
312     }  
313  
314 if (digitalRead(button9) == LOW)  
315 {  
316     while (1)  
317     {  
318         if (digitalRead(button9) == HIGH)  
319         {  
320             inputCode[p] = ('9');  
321             Serial.println("--9--");  
322             delay(1);
```

```

319         ++p;
320         break;
321     }
322 } while (digitalRead(buttonHash) == HIGH);
323 Serial.println("Break*");
324 break;
325 }
326 }
327 }
328 }
329 if (digitalRead(buttonHash) == LOW)
330 {
331     Serial.println("--#--");
332     Serial.println("--P: ");
333     Serial.println(p);
334     Serial.println("--maxIn");
335     Serial.println(maxIN);
336     while (1)
337     {
338         if (digitalRead(buttonHash) == HIGH)
339         {
340             for (i = 0; i < (k); i++)
341             {
342                 if (inputCode[i] == secretCode[i])
343                 {
344                     correct++;
345                 }
346                 //Check Code
347                 Serial.println("InputCode:");
348                 Serial.println(inputCode[i]);
349                 Serial.println("SecretCode:");
350                 Serial.println(secretCode[i]);
351             }
352
353             Serial.println("correct");
354             Serial.println(correct);
355             Serial.println("p");
356             Serial.println(p);
357             Serial.println("k");
358             Serial.println(k);
359
360             reset();
361
362             if ((correct == k) && (p == k))
363             {
364                 accesgranted();
365                 break;
366             }
367             else
368             {
369                 accesdenied();
370                 break;
371             }
372         }
373     }
374 }
375 }
376 }
377 //=====
378
379 _Bool checkid(double idcard) //NFC ID's with Access
380 {
381     Serial.println(idcard);
382     if (idcard == 111111111) //here you need to add the id of your authorised card insted of 1's

```

```
383 {
384     Serial.println("Card1");
385     return true;
386 }
387
388 else if (idcard == 1111111111)
389 {
390     Serial.println("Card2");
391     return true;
392 }
393 else if (idcard == 1111111111)
394 {
395     Serial.println("Card3");
396     return true;
397 }
398 else
399 {
400     Serial.println("NoAccess with this Card");
401     return false;
402 }
403 }
404
405 void accesgranted()
406 {
407     Serial.println("Access granted");
408     delay(1);
409     digitalWrite(ledOpen, HIGH);
410     Serial.println(doorPin);
411     analogWrite(doorPin, 255);
412     delay(2500); //2.5sec
413     digitalWrite(ledOpen, LOW);
414     analogWrite(doorPin, 0);
415 }
416
417 void accesdenied()
418 {
419     Serial.println("Access denied1");
420     delay(1);
421     digitalWrite(ledClose, HIGH);
422     delay(1000); //1sec
423     digitalWrite(ledClose, LOW);
424 }
425
426 void reset() //InputCode-Array mit 0 füllen
427 {
428     Serial.println("Reset");
429     int r; //VarReset
430     for (r = 0; r < (maxIN); ++r)
431     {
432         inputCode[r] = '0';
433     }
434 }
```

C:/Users/Luca/OneDrive/Dokumente/HSR/Git/TuerschlossNFC/tuerschlossnfc.c