
Sorry für diese Art der Darstellung.

So stelle ich sicher, daß niveaulose Trittbrettfahrer die Daten wenigstens selbst einmal abgetippt haben, bevor sie sie unter eigenem Namen veröffentlichen. Wenn jemand die Original-Daten für sich privat nutzen will - ich habe Email.

(c) 2008 Gerhard Günzel

Datei : RFIDSANDWICH.BAS
Datum : 14.06.2008
Letzte Aenderung:

Das RFID-Sandwich ist eine Hardware-Entwicklung von Gerhard Günzel.
http://www.emagu.de/rfid/mit_em4095.htm
Die Software ist auf das Demokit von MCS aufgesetzt. Hier werden keine Taster und LCD verwendet; und zum Programmieren wird der Zugang über einen Mastertag, der selbst einprogrammiert werden kann, ermöglicht.

Beschreibung und Funktionsweise:

Hardware:

RFID-Sandwich ist ein RFID-Leser in runder Form, so dass RFID-Platine und Powerplatine in eine Unterputzdose passen. Sandwich daher, weil die Platinen so konstruiert sind, daß sie übereinander mit Abstandsbolzen verschraubt werden können.
Die Hardware besteht aus Mega8, RFID Leser EM4095 und MAX202 für Serielle. Der Mega8 ist der Programmprozessor, der EM4095 ist der RFID-Leser und der Max 202 ist für die serielle Schnittstelle (8,N,1,19200). Die Serielle wird zur Ausgabe von Klartext während des Programmablaufs benutzt. Es werden nur ASCII Zeichen für ein Terminalprogramm auf dem PC ausgegeben. Zur Status-Anzeige ist eine Led eingebaut, die parallel zur Terminalausgabe verschiedene Zustände optisch am Reader anzeigt. Beschreibung folgt unten.
Schnittstellen zum PC sind SPI (10pol) und RS232 (3pol) als Stiftleisten. Antenne und LED jeweils (2pol)) Stiftleisten.

Generell:

Zugang zum Reader erfolgt über einen Master Tag. Das kann ein xbeliebiger Tag mit 4102 IC sein. Der Master wird programmiert, wenn man einen Tag an den Leser hält und Reset drückt. Wird der Tag 10 mal erkannt, dann wird er gespeichert. Als Anzeige hierfür blinkt die LED im 2 Sekunden Takt 10 mal und blinkt dann 5 sec. lang in schnellem Takt. Wird der Master nun entfernt und nochmal Resetet, bleibt der Tag als Master gespeichert, sperrt aber nicht. Der Master kann dann aber auch programmiert werden. Es ist Platz für 20 Tags im Speicher.

Programmiermodus - Tags einlesen und speichern:

Mastertag einlesen. Led blinkt schnell für 5 Sekunden und wird für zwei sec dunkel und Relais zieht an, wenn Mastertag gespeichert ist. Led leuchtet dann ständig, bis ein neuer Tag dran gehalten wird. Led blinkt dann für 5 Sekunden langsam als Zeichen für gespeicherten Tag. Anschließend leuchtet die Led wieder ständig.

Lesemodus:

Wird ein Tag an den Leser gehalten, der nicht gespeichert ist, dann blinkt die Led im halb-Sekunden Takt. Das ist nur ein Zeichen, daß ein Tag gelesen wurde. Wird ein gespeicherter Tag erkannt, dann blinkt die Led kurz zweimal und wird für zwei Sekunden dunkel. In den zwei Sekunden wird ein Signal an einem Transistor mit Open Kollektor ausgegeben (aktiv Lo). Ist die Powerplatine angeschlossen, so zieht für zwei Sekunden ein Relais.

Programmbeschreibung:

- Beim Programmstart werden die üblichen INIT-Geschichten für den Compiler festgelegt, wie Declare, Dim, Config, Alias, Baud usw.
- Mit "Config Hitag" wird die Lib von Bascom eingebunden.
- Zwei Funktionen werden im Programm deklariert und benutzt
Havetag und Is_it_master. Havetag schaut nach, ob ein Tag schon gespeichert ist gibt entweder 0 oder die Tagnummer im Speicher zurück. Is_it_master testet ob der eingelesene Tag der gespeicherte Master ist.
- Es werden eigentlich nur drei Signale vom EM4095 benutzt: MOD,SHD,DEMODO_OUT
MOD und SHD werden beim Programmstart auf Lo gezogen, so daß der EM4095 als reiner Reader arbeitet. DEMODO_OUT ist das Nutzsinal, das vom Interrupt1 verarbeitet wird.
- Das Auslesen der Tags erfolgt über den Bascombefehl "Readhitag".Es wird die Startadresse des 5-stelligen Arrays übergeben, wo dann die Tag-Daten zu finden sind. Gleichzeitig wird der vorher mit (Config Hitag) festgelegte Interrupt frei gegeben, über den die Daten vom RFID-Chip gelesen werden.
Die genaue Beschreibung ist in der em4095.lib von BASCOM zu finden :-)

Programmablauf:


```

Dim Emastertag(5) As Eram Byte           'Master Tag
Dim Smastertag(5) As Byte               'Master Tag im SRAM im EEPROM
Dim Stags(100) As Byte                 'Speicher im SRAM
Dim Btags As Byte , Tmp1 As Byte , Tmp2 As Byte
Dim K As Byte , Tel As Byte , M As Byte
Dim Mastercount As Byte                'Zähler für Einlesen des Mastertag

Config Hitag = 64 , Type = Em4095 , Demod = Pind.2 , Int = @int0

Print "Tueroeffner"                    'Terminal = ANSI 8bit

On Int0 Checkints Nosave                'INT0 - push und pop in der LIB
Config Int0 = Change                    'Int bei Flankenwechsel
Enable Interrupts                       'Int generell freigeben

'
' nach Reset irgendeinen Tag als Mastertag speichern wenn er 10 x erkannt wird
' LED blinkt im 2 sec Takt
'
Do
  If Readhitag(tags(1)) = 1 Then          'Tag lesen - Lib aufrufen und INT1 ena
    Led = 0                              'Led aus wenn Tag erkannt wird
    For J = 1 To 5
      Print Hex(tags(j)) ; ", " ;        'anzeigen
    Next
    Waitms 2000                          'Für Anzeige
    Led = 1                              'Led wieder aus
    Incr Mastercount                     'Zähler für 10x Mastertag
    If Mastercount = 10 Then
      Print "10x erkannt, speichern"

      If Readhitag(tags(1)) = 1 Then
        For J = 1 To 5
          Print Hex(tags(j)) ; ", "
          Smastertag(j) = Tags(j)
          Emastertag(j) = Tags(j)
        Next
      End If
      Exit Do
    End If
  End If
  If Readhitag(tags(1)) = 0 Then
    Print "nixda" : Exit Do
  End If
Loop

'----- Programmstart
'
'----- Alle 20 Tags aus EEprom lesen und im SRAM speichern
'
For Idx = 1 To 100                       '100 Speicher - 20 Tags zu je 5 Byte
  Stags(idx) = Etags(idx)                'übertragen ins SRAM
  Print Hex(stags(idx)) ; ", " ;
Next
Print

'----- Mastertag lesen und im SRAM speichern
'
Print "Master Tag: " ;                   'Leerzeile nach Speicher auslesen
For Idx = 1 To 5                         'Mastertag
  Smastertag(idx) = Emastertag(idx)      'übertragen ins SRAM
  Print Hex(smastertag(idx)) ; ", " ;
Next
Print                                     'Leerzeile nach Speicher auslesen

'----- Btags setzen - wieviele Tags sind gespeichert
'
Btags = Etagcount                        ' Zahl der gespeicherten Tags
If Btags = 255 Then                      'Leere Speicherzelle - leeres eeprom
  Print "EEPROM leer"                   'nach Programmierung ist 0
  Btags = 0 : Etagcount = Btags         ' Btags nullen und im eeprom speichern
Else                                     'Es sind Tags gespeichert
  For J = 1 To Btags
    Tmp2 = J * 5                         'Endadresse vom Tag
    Tmp1 = Tmp2 - 4                      'Startadresse vom Tag
    Print "RFID ; " ; J                  'Debug, Speichernummer und Tag anzeigen
    For Idx = Tmp1 To Tmp2
      Print Hex(stags(idx)) ; ", " ;
    Next
  Next

```

```

    Print
  Next
End If

'
'----- Abfrage Tag
'
Led = 1 'Led an als Betriebsanzeige
Do
  Print "Lese..."
  If Readhitag(tags(1)) = 1 Then 'Tag lesen - Lib aufrufen und INT1 ena
    Led = 0 'Led aus wenn Tag erkannt wird
    For J = 1 To 5
      Print Hex(tags(j)) ; ","; 'anzeigen
    Next
    Print " erkannt"
    Waitms 200 'Für Anzeige
    Led = 1 'Led wieder aus

'----- zuerst Masterabfrage

M = Is_it_master(tags(1)) ' Call Funktion Mastertag abfragen
If M = 0 Then
  Print "Kein Mastertag erkannt "
End If
If M = 1 Then
  Print "Mastertag erkannt "
  For Tmpl = 1 To 50 'schnelles Blinken der Led für 5 sec.
    Toggle Led
    Waitms 100
  Next
  S3 = 1 'Softschalter setzen zum Speichern
End If 'des nächsten Tag

'----- Ist Tag gespeichert, wenn ja dann Relais

M = Havetag(tags(1)) 'Call Funktion - Ist der Tag gespeichert
If M > 0 Then 'Wenn ja, dann
  Print "Geltiger TAG: " ; M 'Tagnummer anzeigen
  Relay = 1 'Relay an
  Led = 0 'Led aus
  Waitms 2000 'Warte 2 Sec.
  Relay = 0 'Relais aus
  Led = 1 'Led an
Else
  Print "Tag nicht bekannt" 'Tag am Leser nicht gespeichert
End If
Print
Else
  Print "Nichts erkannt" 'Kein Tag am Leser
End If

'----- Speichern der Tags

If S3 = 1 Then 'Softschalter zum Programmieren
  Print "Button 3" 'Taste3 für Tag steichern
  Print "Add RFID" 'Anzeige
  Do
    If Readhitag(tags(1)) = 1 Then 'INT1an
      If Havetag(tags(1)) = 0 Then 'Bisher nicht gespeichert
        If Btags < 20 Then 'Wenn er noch in Speicher passt,
          Incr Btags 'dann speichern
          Etagcount = Btags
          Idx = Btags * 5 'Offset
          Idx = Idx - 4
          Lowerline
          For J = 1 To 5
            Lcd Hex(tags(j)) ; ","
            Stags(idx) = Tags(j)
            Etags(idx) = Tags(j)
            Incr Idx
          Next
          Cls
          Print "TAG gespeichert" : Waitms 1000
          For Tmpl = 1 To 20 'langsames Blinken der Led für 5 sec.
            Toggle Led 'Ende von Speichern
            Waitms 300
          Next
          Led = 1 'Led an als Betriebsanzeige
        End If
      End If
    End If
  End If

```

```

        Exit Do
    End If
Loop
End If
' If S2 = 0 Then
'   Print "Button 2"
' End If
' If S1 = 0 Then
'   Print "Button 1"
' End If
S3 = 0
Waitms 500
Loop

'-----Function Havetag
'check ob ein Tag schon gespeichert ist
'return 0 wenn nicht gespeichert
'return Speicherplatz 1-20 wenn Tag schon gespeichert ist
Function Havetag(b As Byte ) As Byte
Print "Pruefe ob TAG gespeichert ist: ";
For K = 1 To 5
    Print Hex(b(k)) ; ",";
Next
Print

For K = 1 To 20
    Tmp2 = K * 5
    Tmp1 = Tmp2 - 4
    Tel = 0
    For Idx = Tmp1 To Tmp2
        Incr Tel
        If Stags(idk) <> B(tel) Then
            Exit For
        End If
    Next

    If Tel = 5 Then
        Print "Tag erkannt"
        Havetag = K
        Exit Function
    End If
Next
Havetag = 0

End Function

'-----Function Is it master
'Abfrage ob der gelesene Tag als Master gespeichert ist
'Return 0 wenn nicht Master 1 wenn Master
Function Is_it_master(iim As Byte) As Byte
Print "Pruefe ob TAG Master ist: ";
For K = 1 To 5
    Print Hex(iim(k)) ; ",";
Next
Print

Tel = 0
For Idx = 1 To 5
    Incr Tel
    If Smastertag(idk) <> Iim(tel) Then
        Exit For
    End If
Next

If Tel = 5 Then
    Is_it_master = 1
    Exit Function
End If
Is_it_master = 0

End Function

'-----Interruptroutine
Checkints:
Call _checkhitag
Return

```