Time Stamped On Site Reprogrammable Beacon Keyer

Andy Talbot G4JNT March 2012

The Timestamp Hash

A couple of years ago I proposed a scheme whereby a beacon could generate a unique three character Hash of the date and time. The hash generated is a cryptologically generated combination of the date and time and a 64 bit key unique to a user / beacon. Whilst appearing random, and changing every minute, a beacon keepr who knows the password can verify reception reports by checking the reported hash code against the one that should have been generated knowing the tinme date and key.

To assist logging the hash is made up of a pronounceable triad of letters consisting of consonant-vowel-consonant. ('Y' is treated as a vowel) More details including the Hash algorithm internals can be found at http://www.g4jnt.com/BeaconTimestamp.pdf . Software to generate hash versus date / time lookup tables, and a hash generator from a easy to remember password can be downloaded here http://www.g4jnt.com/BCTMSTMP.ZIP

Beacon Keyer

The module described here is functionally similar to my standard Beacon Keyer module, http://www.g4jnt.com/BcnKeyer.htm but takes in NMEA data from a timing generator such as a GPS receiver and reads the ***$GPRMC*** string to generate the three character Hash in real time.. The Triad, along with the single numerical digit corresponding to the unit of the minutes is sent in CW at the end of the stored message.

The beacon keyer is a small module that generates pre-stored CW messages and controls a Tx / Rx line. As well as CW messages, various delays of up to eight different lengths can be embedded within the message and the state of the Tx/Rx and Key lines can be specifically set for each separate delay period. CW speed can be changed at any point within the message using embedded commands with up to eight different speeds.

The module is programmable on-site for new messages and formats using ASCII commands on a serial interface. Messages are stored in non-volatile memory. As well as message changes, the unique Key data is entered as Hexadecimal values using the serial interface.

Hardware

The standard keyer module is based around a 16F628A PIC microcontroller. The circuit diagram is shown in Figure 1 below. Two active low outputs are provided; a key output carries the CW information, and the Tx output allows a transmitter to be keyed from data stored within the message. A small PCB contains the chip and peripheral components, with power and output connections made via l pads. An LED on the standard version allows the CW message and certain items of programming status to be observed.

A four-way header serves a dual purpose. It allows connection of the serial programming interface which connects directly to the RS232 port of a computer running a terminal programme. One current limiting resistor whose value is uncritical is needed in the TXD line from the PC. It is unlikely any damage will occur if this resistor is omitted, but its inclusion is good practice to limit input current into or out of the PIC pin. It also serves as the in-circuit programming interface for the PIC.

Programming

Programming mode is entered when the RS232 interface is conencted and the module turned on or reset. It detects the PORTB,7 or TXD line being pulled low by the interface.

Programming Summary

Terminal (eg. Hypertrm) set to 1200 baud, N81, full duplex.
Connect RS232 lead, and switch on or reset the beacon keyer which
responds with intro message and menu - "Display / Enter / Send /Key"
The LED will illuminate.

Press D To see current message stored in EEPROM
S To go to normal sending and leave programming mode
E To enter and store new message
K To enter and store a new 64 bit key

Type ***E*** and the prompt for the CW Message appears.

Enter the CW and delay message data, ***[rtn]*** completes the message
The LED will flicker slightly as data is entered

CW Speed and delays are set by entering a special code in the CW string:
<Wx> Where x = A to H, sets CW Speed according to the table below.
<Dxyz> Sets a programmable delay
x = R / T sets Tx/Rx line to Receive or Transmit
y = D / U sets key Down / Up
z = A – H Delays for a duration according to the table below

eg. <WC>G4JNT <WE>G4JNT IO90IV58 <DTDC> sends callsign at 10WPM + space,
then at 15WPM followed by locator, then a delay in transmit mode with the key down for 20 seconds.
If no WPM code is included, a default speed of 12 WPM is used.

The backspace key works for normal character entry. Complete token entries can be deleted by pressing [BkSp] only after the closing >

Up to 128 memory location are available message storage.
A warning is issued when entry overflows the storage capability.
If a mistake is made, press [rtn] (+[rtn]) to go back to D / E / S menu,
then start again with the [E]nter option

Letter Code	A	B	C	D	E	F	G	H	
CW Speeds	6	8	10	12	15	20	24	24	WPM
Delays	1	5	10	15	20	30	60	90	seconds

During sending, the LED shows CW characters, and is on when in Tx with Key Down during delays,
LED is on during programming mode and flickers very slightly during RS232 data entry.
There is an extended off period during Token entry.

To enter a new Key, at the prompt Type ***K*** and the module will reply with
New Key data as 16 Hex characters - no spaces ***0123456789ABCDEF***

No [rtn] is needed, the key is automatically updated after the 16th character is entered

0123456789ABCDEF

Typical screen view during programming input

```
G4JNT Beacon Keyer with Hashed Timestamp
Display / Enter / Send / Key        D

CW  "<WC>TEST DE G4JNT <DRUB>"
Key 01 23 45 67 89 AB CD EF

Display / Enter / Send / Key        E
Enter CW & Delay Data
Only Complete tokens can be erased
<WD><DRUB>TEST DE G4JNT [cr]

 CW  "<WD><DRUB>TEST DE G4JNT"
Key 01 23 45 67 89 AB CD EF

Display / Enter / Send / Key         K
New Key data as 16 Hex characters - no spaces
0123456789ABCDEF
```

Key Generation

The utility ***MAKEKEY.EXE*** generates a 64 bit key from a more easily remembered password. The output is given as 16 hexadecimal characters.

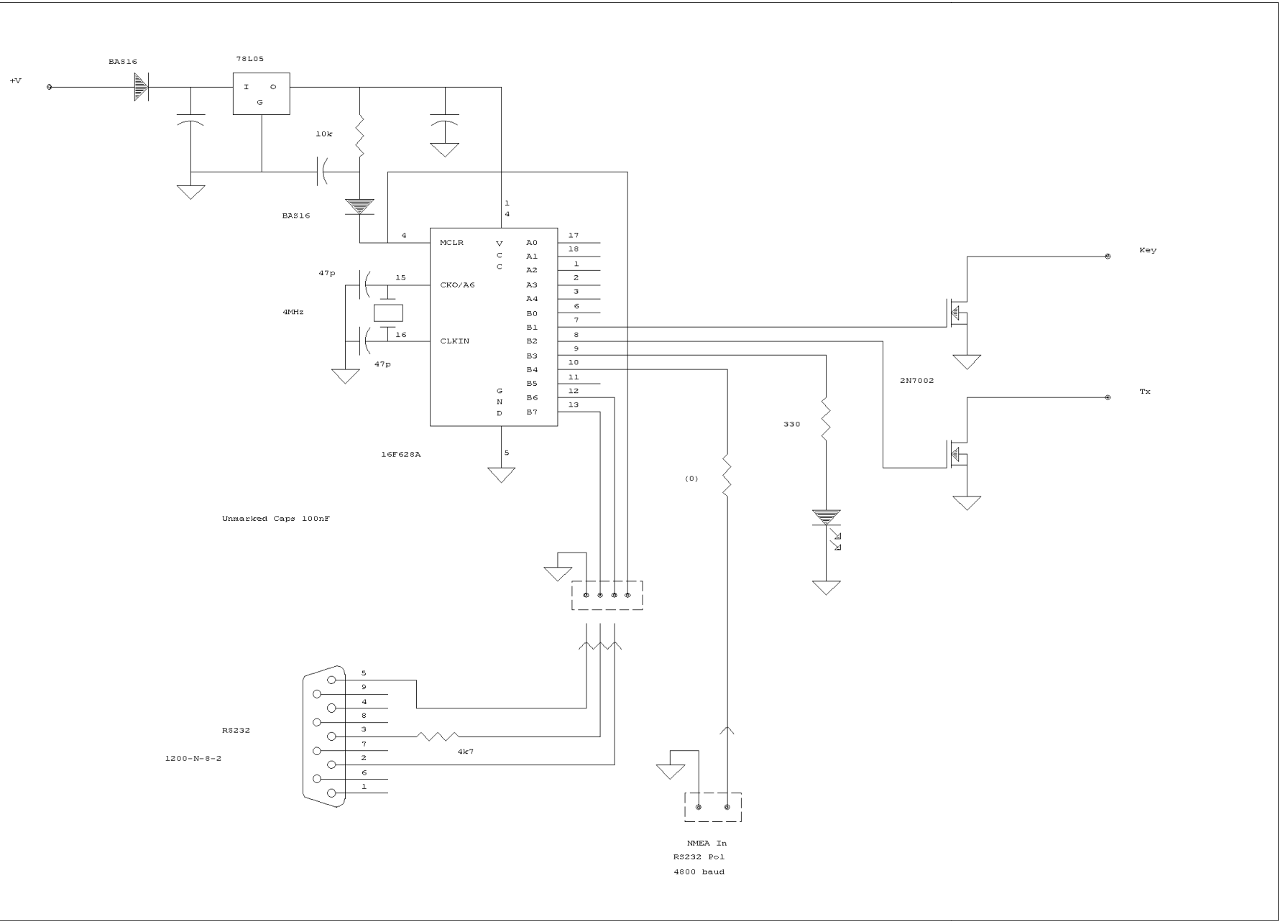
TIMECODE.EXE takes the 64 bit key as input, with the date , and generates a file named TIMECODE.TXT containing all timestamps for each minute of that particular day. The key may be entered either as Hex characters, or via the same password generating process as *MAKEKEY*

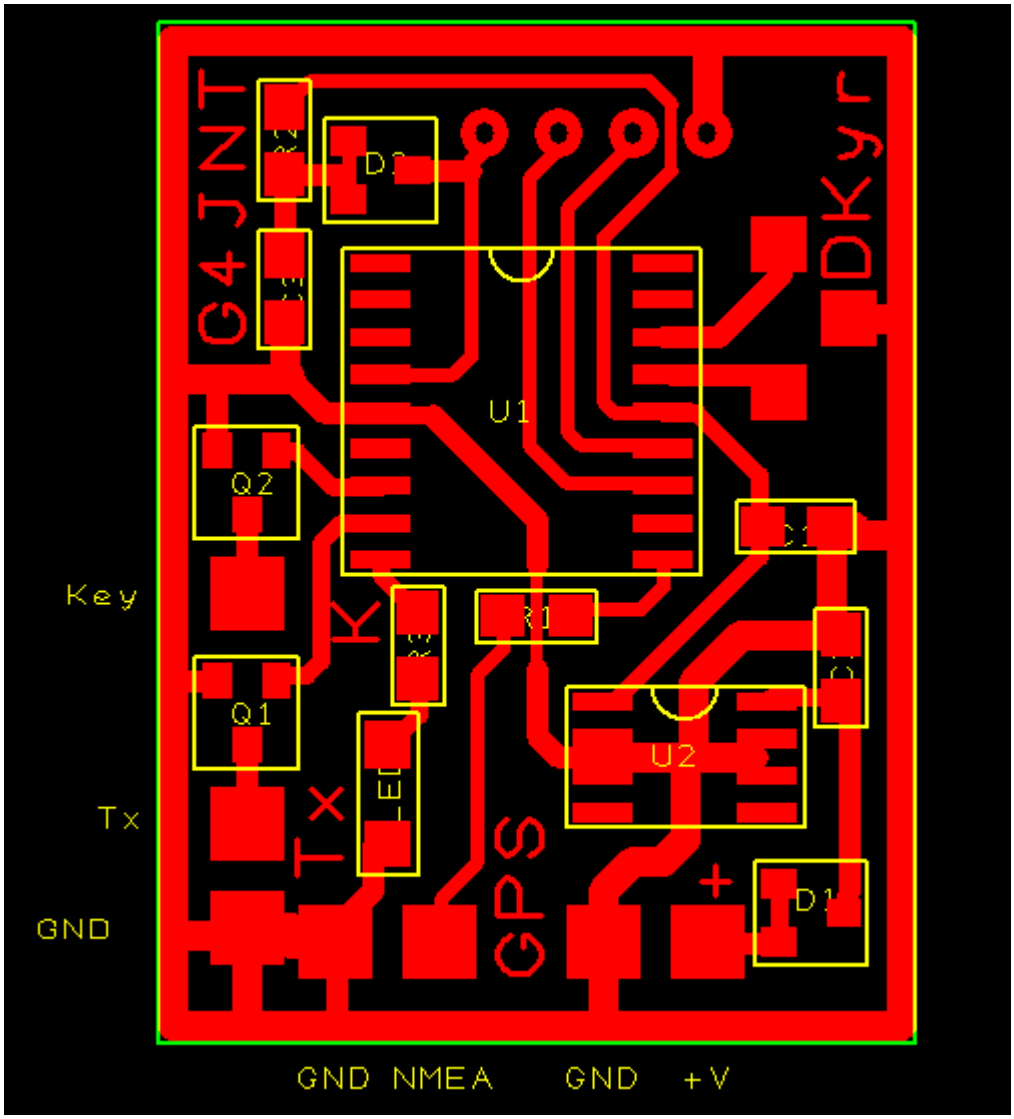
The programme also allows any arbitrary time to be entered and generates the Hash for that particualr time to assist checking received reports.

Both utilities can be found in ***BCTMSTMP.ZIP***

GPS Data Format

Please note, only the GPS Recommended Minimum Sentence ***$GPRMC*** format is decoded. Make sure your GPS receiver or other timing source is set to generate this string


+V
BAS16
78L05
I
O
G
10k
BAS16
1
4
4
MCLR
V
C
C
CKO/A6
CLKIN
G
N
D
A0
A1
A2
A3
A4
B0
B1
B2
B3
B4
B5
B6
B7
17
18
1
2
3
6
7
8
9
10
11
12
13
47p
15
4MHz
16
47p
5
16F628A
Unmarked Caps 100nF
Key
2N7002
Tx
330
(0)
5
9
4
8
3
7
2
6
1
RS232
1200-N-8-2
4k7
NMEA In
RS232 Pol
4800 baud


G4JNT
DKyr
U1
Q2
Key
Q1
Tx
GND
TX
GPS
U2
+
GND NMEA GND +V