HAMgadget description

I built the HAM gadget, giving me the temperature in my shack, and from the GPS satelite(s): UTC time, date, latitude and longitude.
Derived values are: local time (summer / winter) and the Maidenhead locator code (calculated from the GPS coordinates).
'Power saving' can be set with on and off hours (mine are set to 07:00 and 19:00 respectively). Power saving times are handled against your local time, be it summer- or wintertime.

Startup sequence

When switched on a question is asked to retrieve the settings from permanent EEPROM memory. Retrieving the settings is the default. The first time you switch on, there are no settings to be retrieved of course and you have to enter them yourself. Afte you have set them to your values, go to the settings menu and store them manually; they will be retrieved next time.

After that question the credits screen is displayed for 3 seconds and after that the main screen is displayed. The main screen will be empty at first and the top line reads "gps" in lower case, indicating no valid GPS packet has been received yet. Date, time, latitude and longitude will be empty at this point. When a temperature sensor is connected the temperature is the first to be displayed.

When the GPS satellites have been received, the top line is going to read "GPS" in upper case to indicate that a valid packet has been received. How long this takes is strongly dependent on your location: indoors, outdoors, open sky, interference from (wet) trees, etc.

The first thing that will be decoded is the date/time information. For some reason the reception / decoding of the proper coordinates takes a while longer.

For operation, menu structure and hardware used, read the next page(s).

HAMgadget description

Menu structure

On the main screen there are several option that can be activated through the keyboard. This can be a little tricky, sometimes the buttons do not react immediately…

Main screen

The active keys on the main screen are:
- A - Toggle the screen backlight off / on
- B - Toggle between Date/Time display and Lat/Long display
- * - Go to main menu

Main menu

Options:
- 1. Go to Time menu
- 2. Go to Settings menu
- 3. Go to Adjust PowerSave screen
- 4. Display Credits
- #. Go back to Main screen

Time menu

Options:
- 1. Switch to displaying UTC time
- 2. Switch to displaying Local Wintertime
- 3. Switch to displaying Local Summertime
- 4. Go to Adjust Offsets screen
- #. Go back to Main menu

Adjust Offesets screen

Options:
- 1. Adjust wintertime offset
- 2. Adjust summertime offset
- #. Go back to Time menu

Settings menu

Options:
- 1. Show current settings
- 2. Store settings in EEPROM memory
- 3. Retrieve settings from EEPROM memory
- #. Go back to Main menu

Adjust PowerSave screen

Options:
- 1. Adjust hour to switch the backlight on
- 2. Adjust hour to switch the backlight off
- #. Go back to Time menu

Input values

Values, when asked for, can be entered on the keyboard. Press * for a minus sign, press \# to enter the value.

HAMgadget description

Hardware used

The hardware used for this HAMgadget is listed here.

The CPU:

ATMEGA328P (this is also the core of the Arduino UNO). I like to prototype my builds using the Arduino UNO development board, and, when building the and-result, I use a separate ATMEGA328P chip with a 16 MHz crystal and a 10K resistor (as pull-up on pin 1 - the reset pin).

Display:

As a display I chose a 20 x 4 LCD display with an I2C interface. This display is connected through an I2C interface (a two wire interface). The address of these displays can be set on the I2C 'backpack' on the backside of the displays, by soldering little wires to connect two solder pads.
In the picture on the right you can see the small wire I soldered over them, just to the right of the blue box.
The default address of these displays (the ones I got anyway) is 0x27, which is what I used.

Temperature sensor:

For measuring the temperature I used a DS18B20. It operates on a one-wire interface.

GPS receiver:

To receive the GPS signals I use a receiver, VMA430 (U-BLOX NEO-7M). This module has a serial interface. Because the default serial interface was already in use for the temperature sensor we had to use a software serial library to get this one to work.

Keyboard

I used a simple 4 x 4 matrix keyboard for this project. It has 8 connections that have to be connected to the ATMEGA. See the code for how this works.