

LittleGP-30

Jürgen Müller, juergen@e-basteln.de

Document rev. 1.0

2nd November 2016

This document describes the original “pocket calculator” version of the LittleGP-30.

If you have the newer “wide panel” version, please see www.e-basteln.de/lgp30 for documentation.

Contents

| | |
|---|----|
| What’s this? | 2 |
| License terms and credits | 2 |
| Building the Replica | 4 |
| What you will need | 4 |
| Populating the circuit board | 5 |
| Power supply options | 8 |
| Configuring the FPGA | 8 |
| FPGA development | 9 |
| Using the Replica | 10 |
| The real LGP-30 | 10 |
| Getting Started | 10 |
| Control panel | 11 |
| LCD display | 12 |
| Serial terminal | 13 |
| Drum memory display (HDMI) – optional | 15 |
| Drum memory storage (XModem) – optional | 18 |
| Oscilloscope connection – optional | 19 |
| Appendix | 21 |
| Download links | 21 |
| Schematics | 22 |
| PCB layout | 24 |
| Bill of materials | 25 |

What's this?

The LGP-30 was a commercial computer, released in 1956. Due to its simple design and relatively low cost, it may be seen as the first “personal computer” – to be used by a single user as their “desk computer”. (It could sit by your desk, and happened to be the size of a desk.) Designed in the age of vacuum tubes, it needed only 113 tubes in total, of which only 24 were used in the CPU itself! This simplicity was achieved by a bit-serial CPU design, which was tightly integrated with the magnetic drum storage unit. The magnetic drum contained not only the main memory, but also the CPU’s three 32-bit registers, and several tracks with timing signals to control the instruction decoding and execution.

This document describes a replica that is true to the bit-serial implementation and its timing, but uses modern components. The CPU and the magnetic drum storage are recreated in an FPGA – by implementing the complete logic equations published by the LGP-30’s inventor (in a scientific paper and in the computer’s service manual). The magnetic drum, while implemented in on-board memory inside the FPGA, is made tangible via an optional video display of its contents.

This way, one can play with all the quirks of the LGP-30, including the timing behavior of programs, which depends critically on the position of instructions and data on the magnetic drum. For those who want to really dive into the details of the bit-serial design, the clock rate can be slowed down (all the way to single step, bit-by-bit clocking), and the contents of the CPU’s tube-based flip-flops can be inspected on an LCD (all 15 bits of them!).



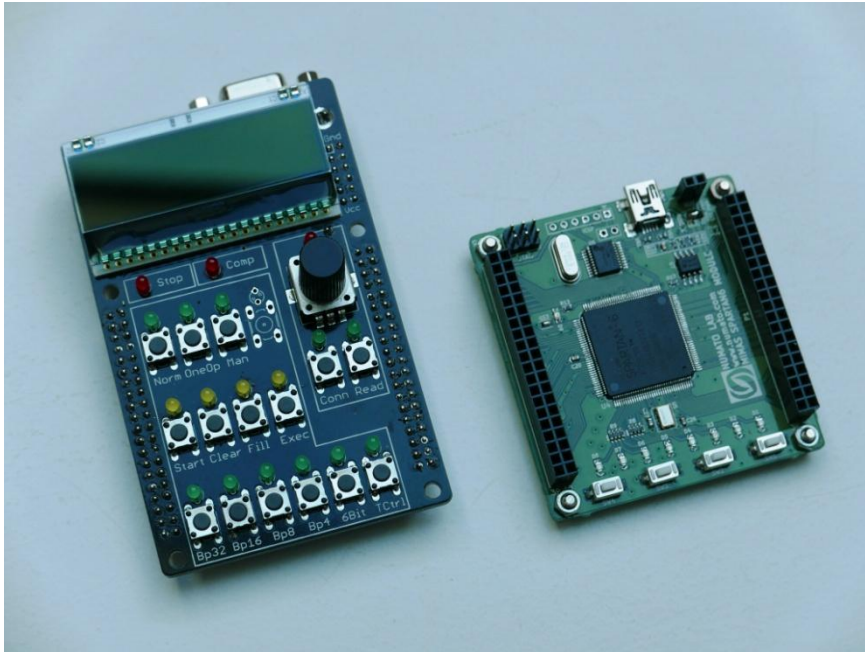
License terms and credits

This material, the PCB layout and the FPGA code are Copyright © 2016 Jürgen Müller, juergen@e-basteln.de. Use for commercial purposes requires my explicit permission in writing. Use for non-commercial purposes is granted free of charge, but I ask that you disclose the source and include this copyright notice. I would appreciate if you let me know when you re-use this stuff.

Use of the following 3rd party components is gratefully acknowledged. These may bring their own license restrictions; please check before you publish this code or use it commercially:

- Minimal DVI Encoder framework by Mike Field, http://hamsterworks.co.nz/mediawiki/index.php?title=Minimal_DVI-D
- TMDS Encoder by Jean P. Nicolle, <http://www.fpga4fun.com/HDMI.html>

- Serial UART by Ken Chapman, Xilinx Ltd,
http://ohm.bu.edu/~dean/Xilinx/KCPSM6_Release7_30Sept13/UART_and_PicoTerm/UART6_User_Guide_and_Reference_Designs_29March13.pdf
- LCD font by Benedikt K., <http://www.mikrocontroller.net/topic/54860>

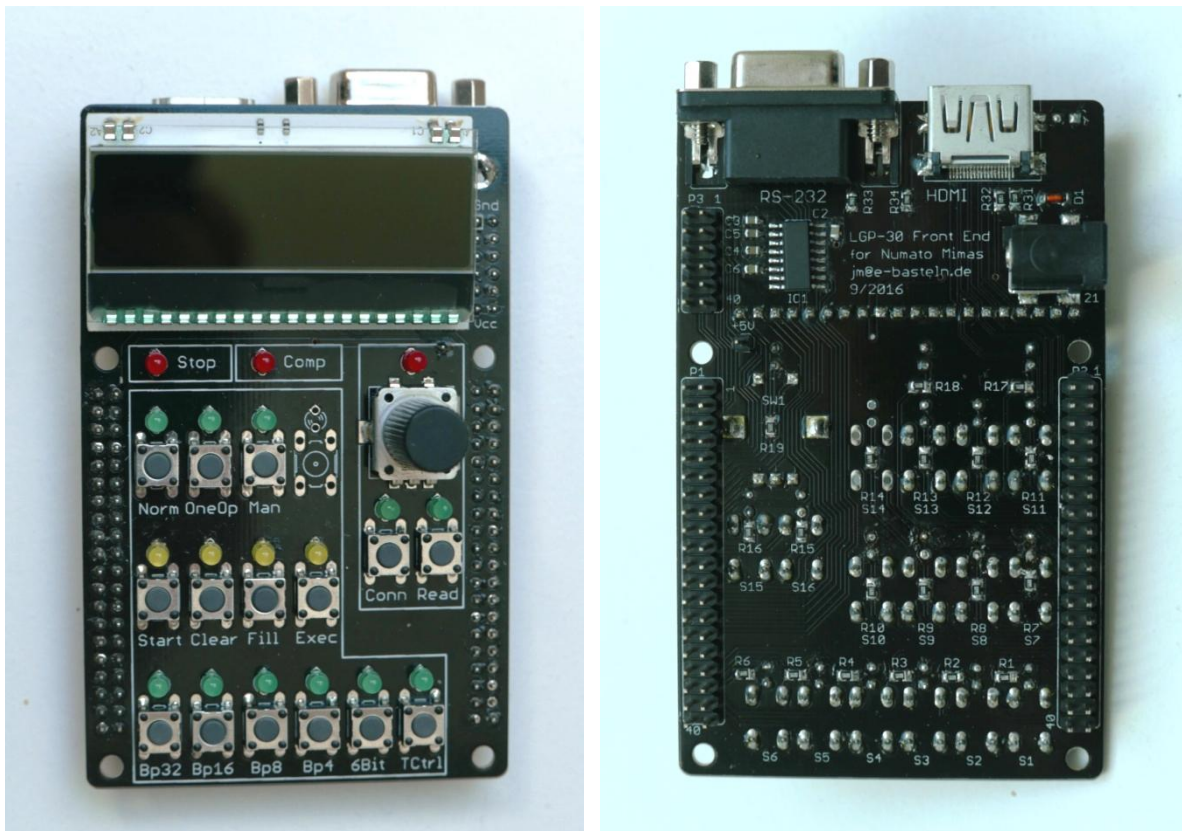


The populated front-end board with the Numato Mimas FPGA board which does the heavy lifting.

Building the Replica

What you will need

- Numato “Mimas” Spartan 6 FPGA board
- Custom PCB and electronic components, see appendix
- Cables:
 - Serial connection: 9-pin Sub-D male to female connection, wired straight through (when connecting to a standard PC RS-232 port, with 9 pin male output)
 - USB connection (to configure the FPGA) and power supply: Mini-USB type B, angled connector, to regular USB type A.
 - HDMI, length < 1.5m recommended
- Hand soldering tools
 - Soldering iron (pen or chisel tip), solder (resin core, 0.5mm diameter or less), wire cutters, tweezers etc.
 - Flux (e.g. pen or gel). I normally don't use extra flux and just rely on the resin core in the solder, but flux is definitely needed for soldering the HDMI connector!



Here's the goal – top and bottom side of the populated board

Populating the circuit board

Most soldering on this board is straightforward. Even if you have not soldered SMD components before, the discrete components (Rs and Cs), as well as the SOIC (MAX3232), should not present a problem. Soldering the HDMI connector can be a bit hairy, however; please see the hints below.

I recommend populating the board in the order of the following sections.

1. HDMI connector

I recommend soldering this connector first, while no other components can get in the way. The challenge is to avoid solder bridges between the connector's pins (especially in the upper part, away from the PCB). The recommended approach is to first tin the pads on the PCB, then place and solder the connector without adding more solder.

- Put flux on the PCB pads. (Needed! There is no solder mask between the pads, and the solder *will* create bridges between them if no flux is applied.)
- Tin the PCB pads. If solder bridges are created, they are removed relatively easily with solder-wick or a suction pump. If the solder does not want to flow towards the pads, add a bit more flux. – The small pads, especially those without a trace connected to them, lift off the PCB quite easily. Hence, be careful and don't "scrub" with the solder-wick or pump.
- Put some more flux on the tinned pads and on the HDMI connector's pins.
- Position the HDMI connector. The two through-holes register with the connector's plastic pins. I recommend clamping the connector down with a small clamp or clothes pin. Check the alignment of pins and pads with a magnifier, and fine-adjust. Do not solder the four mounting tabs yet, in case you need to unsolder the whole connector in the subsequent step.
- Solder each pin individually, with a fine-tipped iron. Try to avoid creating solder bridges, by carefully centering the tip with each pad/pin. If a solder bridge forms, try "pulling" it away from the connector with the tip of the iron. Alternatively, you can carefully apply solder-wick to the pins and pads. – If you have access to a hot-air soldering station, this should work well for this step.

Don't move the tip or the solder wick too far in – it should not touch the pins where they angle upwards, away from the PCB! Solder bridges that form between the pins in this region are very stubborn.

2. SMD IC

The SOIC package of the MAX3232 is still large enough to be soldered pin by pin. If you have not done this before – I like the approach shown in the following video:

<https://www.youtube.com/watch?v=fq8A95AQFYU>

3. Discrete SMD components

The resistors and capacitors are all in 0805 format, which is large enough for conventional hand soldering. Please note that several capacitors are on the top side of the board (where they will later be hidden underneath the LDC). Once again, here's a video showing the soldering steps:

<https://www.youtube.com/watch?v=IrDyUj7ZfVI>.

The revision 2 board has one diode (D1, near the HDMI connector). Soldering technique is the same as for the resistors and capacitors, but please observe the diode's polarity – the band on the package must match the white stripe on the silkscreen.

4. LEDs

LED polarity matters. The flat side of the plastic package must match the footprint shown on the silkscreen. If this is hard to see for your LEDs, observe the different lengths of the LED's pins: Make sure that the long leg is always oriented towards the switch which corresponds to the respective LED.

The LEDs look best if they are nicely aligned. I recommend inserting the LEDs with straight legs, soldering only one leg first, and adjusting/resoldering them if needed before soldering the second pin.

Recommended LED colors:

- | | | |
|--------------------|--------|---|
| ▪ LED 1 - 6, 11-16 | green | show on/off status of toggle switches |
| ▪ LED 7 - 10 | yellow | show that an "action" switch is enabled |
| ▪ LED 17 - 19 | red | status indicators |

Note that one LED and switch (LED14 and SW14) are spares which do not currently serve a function in the LGP-30 replica. You can leave them unpopulated if you prefer.

5. Switches & Encoder

The switches, when pushed firmly into the PCB, should register precisely and can be soldered without further alignment. They can be inserted in two orientations which both work. For the encoder, soldering the two mounting tabs is recommended for mechanical stability.

A small knob will help in handling the encoder. The one on the parts list (see appendix) is cheap and unobtrusive, but needs to be drilled out if you don't want it to sit very high: 7 mm diameter drill, about 6 mm deep. Wrap the knob with a few turns of masking tape before clamping it in a vise – the plastic is quite soft.

6. Pin headers & socket strips

The two large pin headers P1 and P2 (2*20 pins each) need to be populated on the *bottom* side of the PCB, to mate with the Numato Mimas board. The same is true for the single +5V supply pin (near the encoder's LED – present on revision 2 PCBs only.)

The 2*6 pin header P3 is optional. Depending on the FPGA configuration, it may be used to output test signals for debugging, or vertical and trigger signals for an analog oscilloscope display. (See the oscilloscope section in the "Using the replica" chapter for details.) I recommend leaving P3 unpopulated unless you have a need for it. If used, it is meant to be populated on the *bottom* side of the PCB too.

The LCD should be socketed, using precision (turned) socket strips for good mechanical contact:

- The two LED supply contacts in the back (2 pins each) *must* be socketed, because one of them is located above the RS-232 connector. Soldering in the complete LCD assembly would prevent soldering of the RS-232 connector, and vice versa. **Mount these socket pins before**

installing the RS-232 connector! Clip the pins of this socket flush with the PCB before soldering, so they do not interfere with the RS-232 connector in the following step.

- The 20 pins in the front can alternatively be soldered without a socket, to place the LCD at a slight angle. But I recommend using a socket strip as well, in case the LCD needs to be removed for troubleshooting.

7. RS-232 connector

RS-232 connectors come in two slightly different mechanical packages. This PCB layout is meant for the U.S. flavor, with a distance of 7.2mm between the connector's front edge (PCB edge) and the first row of pins.

The pin numbering is for a female Sub-D 9 pin connector. This will be connected 1:1 to the male connector found in PCs, for use with a PC-based terminal emulator program.

Make sure that you have the two socket pins for the LCD (located on the top side, opposite the RS-232 connector) populated first! See step 6.

8. LC Display

The DOGM-132 displays (132*32 pixel graphical LCDs) come in different versions. The recommended EA DOGM132S-5 is optimized for inverted operation, with bright pixels on black background, and meant to be used with a backlight. EA LED55X31-G is a yellow/green backlight, which results in a color scheme similar to the original LGP-30 oscilloscope. Other models of the DOGM-132 series, including transfective versions without a backlight, can be used alternatively.

When using a display with backlight:

- Remove the protective foils from the display back and backlight front.
- Insert all display pins into the backlight's corresponding holes.
- Install the LCD+backlight package into the sockets soldered previously.
- Solder the 2*2 LED supply pins from the top, to electrically connect the backlight board to these pins.

9. Visual inspection

Congratulations – your LGP-30 replica is fully populated! Before you move on to review the power supply options in the next section, I recommend a thorough visual check of your work so far:

- Have all SMD components been populated, and with the right values?
- Is the polarity of all LEDs correct? Besides looking for the flat side plastic body, you can also see the electrodes inside the slightly turbid plastic. The larger electrode is the negative one, and should be on the side away from the push button.
- Any solder bridges? Inspect the MAX3232 IC and the HDMI connector in particular. The headers P1, P2 and P3, as well as the sockets for the LCD, also deserve a closer look.
- Have all pins actually been soldered? It is easy to miss some of the push-buttons' pins, or some of the pins on the large headers. In addition to malfunctions right from the start, this can lead to annoying problems with intermittent connections later on.

You will be able to follow up with a functional test once you have configured the FPGA – please see the respective section below.

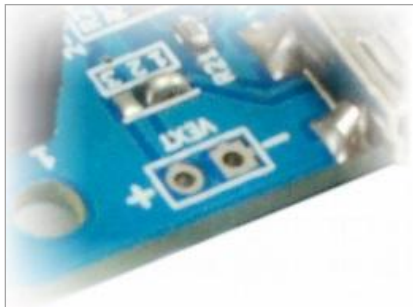
Power supply options

The Mimas board and front-end PCB require a +5V regulated supply, with a current rating of 300 mA or more. I have measured 240..260 mA current consumption with an HDMI monitor connected, and ~40 mA less without a monitor. Any regular USB port or small USB-style power supply should be adequate.

The following discussion only applies if you use revision 2 of the LGP-30 front end PCB (black PCB, with DC power jack on board, dated 9/2016). This revision of the PCB has a connection to the +5V supply on the Numato Mimas board, which is used to provide +5V on the HDMI output. It also has the option to supply power to both boards via the more easily accessible coaxial connector on the piggyback board (rather than the USB connector on the Mimas board).

To use these features, make the following changes on the Numato Mimas board:

- Populate the “VEXT” connector on the Numato Mimas board with a 2-pin female header. This will mate with the single +5V pin on the piggyback board; the ground connection will be established via the two large 40-pin connectors.



- Configure the power selection solder jumper (right behind the VEXT connector). Two options here, depending on your personal preference:
 - Connect pins 2 and 3 only: Power will always have to be supplied via the DC jack on the piggyback board, even when a USB connection for programming is made.
 - Connect all three pins: Power can be supplied either via USB or via the DC jack. More convenient, but some attention is required: You must *not* supply power via both connections at the same time! This would result in the two power supplies fighting over the definition of “+5V”, and could potentially damage a power supply.

If you don’t want to use the +5V option at all, or are using a revision 1 LGP-30 PCB, you can leave VEXT unpopulated, leave the solder jumper in the 1-2 position, and always supply power to the boards via the Mimas USB connector. In this case, you can also leave the DC power jack unpopulated. Flat-screen TVs and some (mostly older) monitors may however not recognize the HDMI output in this configuration.

Configuring the FPGA

To configure the FPGA with an available .BIN binary file, it is *not* necessary to install the Xilinx development environment. (See the next section for details on that package.)

- Obtain the BIN file for this project (see appendix).

- Download the FPGA Configuration Tool, mimasConfig.exe, from Numato:
<http://productdata.numato.com/assets/downloads/fpga/mimas/mimasConfig.exe>
- Download the USB Driver from Numato. Unpack the ZIP file, but don't install it yet:
http://productdata.numato.com/assets/downloads/common/numato_lab_usb_cdc_driver.zip
- Connect the Numato board to the PC via a USB cable. You can do this either with the bare Mimas board, or with the freshly soldered piggyback board already plugged in.
- When prompted on the PC, install the USB driver.
- Run mimasConfig.exe, which does not require installation. Select the correct COM port (check in the device manager if unsure) and binary file, and click "Program".
- After programming is complete, the FPGA will automatically start up. You should see the LCD light up and display its traces and register letters, similar to the picture on page 2. Several LEDs should also light up, and you can toggle them by pressing the mode switches (Normal/One Op/Manual) and the toggle switches in the bottom row.
- If this is the case – congratulations, your LGP-30 has just seen "first light"! You are ready to move on to the "Using the Replica" section. If you don't get the expected behavior, I recommend going back to another careful visual inspection as the first step of troubleshooting.
- Documentation from Numato can be found at <https://docs.numato.com/doc/mimas-spartan-6-fpga-development-board/>

FPGA development

If you want to dive into modifying the FPGA code, free development tools for the Spartan-6 FPGA family are available from Xilinx. They are a bit unwieldy, however:

- A Xilinx Account is needed to download the tools. Free registration under <https://secure.xilinx.com/webreg/createUser.do>
- Download Xilinx ISE Webpack (a 6 GByte download!)
https://secure.xilinx.com/webreg/register.do?group=dlc&htmlfile=&emailFile=&cancellink=&eFrom=&eSubject=&version=14.7&akdm=1&filename=Xilinx_ISE_DS_Win_14.7_1015_1.tar
- Xilinx ISE is no longer maintained and is not fully compatible with Win8 and Win10. The License Manager and Project Navigator both just close when you try to open a file. This can be fixed as follows, quoted from <http://www.eevblog.com/forum/microcontrollers/guide-getting-xilinx-ise-to-work-with-windows-8-64-bit/>:
 - Fixing Project Navigator, iMPACT and License Manager.
Note: I am assuming you are using ISE 14.7 and have installed it to the default location
 - Open the following directory: C:\Xilinx\14.7\ISE_DS\ISE\lib\nt64
 - Find and rename libPortability.dll to libPortability.dll.orig
 - Make a copy of libPortabilityNOSH.dll (copy and paste it to the same directory) and rename it libPortability.dll
 - Copy libPortabilityNOSH.dll again, but this time navigate to C:\Xilinx\14.7\ISE_DS\common\lib\nt64 and paste it there
 - In C:\Xilinx\14.7\ISE_DS\common\lib\nt64 Find and rename libPortability.dll to libPortability.dll.orig
 - Rename libPortabilityNOSH.dll to libPortability.dll
- A brief tutorial for first steps with Xilinx ISE: <http://langster1980.blogspot.de/2014/10/elbert-v2-fpga-tutorial.html>

Using the Replica

The real LGP-30

If you have decided to build this kit, you have probably familiarized yourself with the design and operation of the original LGP-30 to some extent. Several nice web pages with LGP-30 descriptions, technical documentation, and reminiscences of the early days of programming are easily found online.

The University of Stuttgart hosts the most comprehensive site, with technical documentation, a software collection on ASCII-converted paper tapes, and a description of their own, working LGP-30. Please see the Appendix for links.

The LGP-30 Operating Manual is specifically recommended as a starting point for your exploration of the LGP-30: ftp://bitsavers.informatik.uni-stuttgart.de/pdf/royalPrecision/LGP-30/LGP-30_Operations_Manual.pdf

Getting Started

Connecting peripherals

Only two connections to the LGP-30 replica are required:

- Power supply: +5V either via the USB port on the Mimas board which you have already used to configure the FPGA, or (if you have board revision 2) via the DC power jack on the LGP-30 board. See the section on “Power supply options” above. To keep the USB port connected while the LGP-30 front end is installed, an angled USB mini connector is required.
- Serial port: An external ASCII terminal is required to simulate the Flexowriter. See the “Serial terminal” section below.

Further optional connections (HDMI monitor for drum display, oscilloscope for old-style register display) are discussed in the respective sections below.

Loading programs

There are three ways to load programs into the LGP-30 replica:

- Reading “paper tapes”
 - Text files are uploaded via your terminal program to simulate paper tapes.
 - The 10.4 loader program is already preinstalled on the replica’s drum.
 - Press the following control panel buttons to start the loader:
Manual – Clear – Normal – Start
 - Tell your terminal program to send the paper tape text file.
 - Most paper tapes will end with a command that sets the program counter to the start address, so you simply have to press Start once more to start the newly loaded program.
- Loading previously stored drums

- Complete drum contents can be stored and re-loaded if you use a terminal program that supports the old XModem file transfer mode. (1970's technology meets 1950's computer!)
- See the "XModem" section below for details.
- Manual bootstrap from an empty drum
 - The hard way, for "real programmers" ;-)
The papertape for the 10.4 loader begins with an interesting self-extending bootstrap sequence. This is entered in a semi-manual way; then the boot loader can read the main program from the remainder of the tape.
 - To start from scratch and remove the pre-installed 10.4 loader, first load an empty drum via XModem.
 - The bootstrap process is described in the 1963 edition of the Programming Manual. German version available here – see page 27:
ftp://ftp.informatik.uni-stuttgart.de/pub/cm/lgp30/docs/EC13-022_Programmierungsanleitung_Apr1963.pdf

General hint: Paper tape files should *not* have a carriage return (or other characters) after their final "conditional stop" character! When reading a paper tape, this character will remain in the RS232 input buffer and may get in the way of subsequent operations, e.g. an XModem download. Many of the Stuttgart paper tape files do have a final carriage return; I recommend removing it. To flush the buffer of pending characters, switch the LGP-30 to Manual mode and activate "Read" – possibly several times, until no further input is received.

Control panel

LGP-30 Control Panel

All relevant buttons and indicators from the LGP-30 control panel are present. See the LGP-30 operation manual for a detailed explanation of their function.

- Top row: Mode switches (Normal, One Operation, Manual). Pushing one button selects its mode and releases any other mode. The LEDs indicate the active mode.
- Middle row: Operation buttons (Start, Clear Counter, Fill Instruction, Execute Instruction). Pushing a button triggers an operation. Depending on the Mode selection, only some of these buttons are active. The LEDs indicate which buttons are active.
- Bottom row: Toggle switches (Breakpoints, 4/6 bit input, Transfer Control). Each push toggles their state, and the LEDs show the current state.

The LGP-30's power-up, standby and shutdown functionality is not replicated.

Flexowriter controls

The block on the right-hand side has two buttons normally located on the Flexowriter:

- "Connect" switch (toggle): Controls whether the "Start" button on the Flexowriter or a "conditional stop" character read by the Flexowriter are passed to the LGP-30. This should be enabled most of the time, but will need to be disabled e.g. for a semi-manual bootstrap

procedure. (The bootstrap involves reading word for word from the tape, each followed by manual fill/execute instruction commands).

- “Read” toggle: Combines the functions of the Flexowriter’s “Start Read” and “Stop Read” buttons. It will start and stop the paper tape reader – in case of the replica by setting or clearing the “CTS” signal on the RS-232 port.

Encoder

The Encoder in the upper right corner operates in two modes. You can toggle between these modes by pushing down on the encoder button. The active mode is indicated by the LED above the encoder:

- Normal mode (LED off, active after startup): Shifts the HDMI drum display left and right.
- Clock mode (LED on): Controls the LGP-30 CPU clock rate (rotation speed of the drum in bits/second). The standard speed is 125 kHz; it can be increased up to 1.25 MHz and decreased down to 1.25 Hz in three steps per decade.

At speeds of 250 Hz and below, the HDMI drum display will switch to animated mode, showing the drum rotation and the position of the read/write heads. Turning the encoder further left from 1.25 Hz enters single step mode: Each further encoder step to the left will then move the drum by one bit. (Note that this is much slower than the LGP-30 “One Operation” mode, which executes one CPU operation per step.)

Option switches

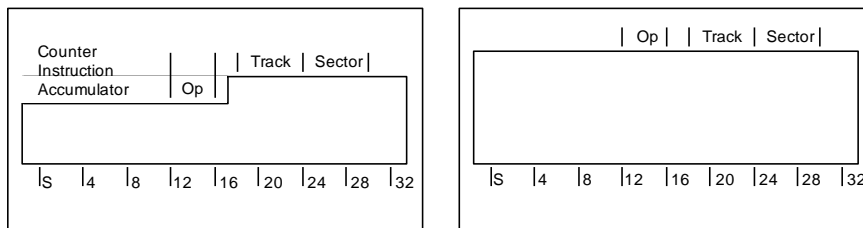
Four switches on the front edge of the Numato Mimas board may be configured to control options of the replica. At this time the following are supported:

- SW4: Disable the extended information in the LC display. True to the original LGP-30, only the CPU register contents will be displayed; flip-flop and clock rate display will be disabled.
- SW3: In RS-232 output, disable the VT-100 ESC sequence for black/red color shifting. This may be useful when using a non VT-100 terminal, or when capturing LGP-30 output to a text file (“punching a tape”).

LCD Display

The LCD’s main task is to replicate the functionality of the oscilloscope display in the LGP-30: It shows the CPU registers – from top to bottom: Program counter C, instruction register R, accumulator A. 32 bits are displayed, including the spacer bit in the rightmost position. The most significant bit (sign bit) is on the left.

In addition the LCD displays the status of the LGP-30’s 15 electronic flip-flops, as well as the current CPU clock rate. These displays were not present on the original LGP-30’s user control panel, although the flip-flop circuit boards had neon bulb indicators for troubleshooting. The additional display items are on by default, but can be toggled off via option switch SW4 on the Mimas board.



Optional: Print and cut out one of these cardboard overlays to read the bit positions on the LCD more easily. The left overlay shows the CPU registers only, the right version leaves room for the clock rate and flip-flop display.

Serial terminal

Settings

An ASCII terminal with serial (RS-232) connection is required to use the LGP-30 replica. It replaces the Flexowriter and its paper tape reader and punch. If a terminal program with XModem capability is used, it can also be employed to store and recall the contents of the magnetic drum memory. (See the separate section below.)

Set the RS-232 parameters to

- 19200 Baud
- 8 data bits
- 1 stop bit
- No parity
- Hardware handshake (RTS/CTS)
- No local echo

The replica translates the Flexowriter 6-bit code to ASCII. Plain ASCII text is used, with one exception: The Flexowriter's "color shift" code, which switches between black or red typewriter ribbon, is translated into VT-100 escape sequences to toggle between normal and boldface characters. Use of these escape sequence can be disabled by option switch SW3 (on the Mimas FPGA board); it is enabled by default.

The physical data rate is chosen quite high, to enable XModem transfers of complete drum contents at reasonable speed. (19200 Baud is the fastest rate handled by real VT-100 terminals, in case you want to connect one.) But the effective transfer speed in regular LGP-30 operation will be much slower: Approx. 200 cps (characters per second) when reading, and 10 cps when printing. This matches the speeds of the high speed tape reader and the Flexowriter, respectively. The slower speeds are required because the CPU timing during input and output operations needs to match the I/O timing. When running the CPU at lower or higher clock rates, I/O speed is adjusted accordingly.

Special characters and control buttons

The ESC key has a special function: It simulates the "Start" key on the Flexowriter, which is identical in function to the "Start" button on the control panel.

Reading a “ ” (conditional stop) character from the keyboard or from a text file also triggers a “Start” signal. In addition, it stops further input by clearing the CTS flow control signal. Input will be re-enabled by the next Input instruction executed the LGP-30.

Two buttons on the replica’s control panel provide controls which were originally located on the Flexowriter:

- The “Read” button, located on the replica’s control panel, is a toggle switch. It replaces the “Stop Read” and “Start Read” switches on the original Flexowriter. It can be used to manually start and stop input from the serial terminal by changing the state of the CTS handshake signal. Note that, different from the real Flexowriter, the replica will not accept input from the keyboard while the “paper tape” is stopped. (Since the CTS signal will of course stop all RS-232 input.)
- The “Connect” button, also on the control panel, represents the “Connect” lever on the Flexowriter. The “Start” signals triggered by the ESC key or the “ ” conditional stop character will only be sent to the logic unit when the Connect switch is on. It should be on in regular operation, but needs to be switched off e.g. for a manual, tape-assisted bootstrap. (See the “Getting started” section.)
- There is no “Conditional Stop” control on the replica. On the real Flexowriter, this can be used to suppress the “stop the tape reader” function of the “ ” character. I have not found any use for this in the replica and assume that it was only needed if one wanted to copy a tape on the Flexowriter. So, the “conditional stop” character is actually an “unconditional stop” on this replica... If you run into the need to disable the stop function, please let me know!

Terminal programs

A real VT-100 terminal should be usable, but I recommend a terminal emulation program on a PC. These can also send and receive text files, to replace the Flexowriter’s paper tape reader and punch. The papertape files available on the internet are directly usable, specifically those from the Stuttgart computer museum: <ftp://ftp.informatik.uni-stuttgart.de/pub/cm/lgp30/papertapes/>

If the terminal program supports the XModem file transfer protocol, this can be used to load and store complete images of the magnetic drum – see the chapter on XModem below.

Under Windows, I like Tera Term, a free program which is still actively maintained: <https://ttssh2.osdn.jp/index.html.en>. If you use Tera Term, insert or edit the following entries in teraterm.ini:

```
; XMODEM receive command
XModemRcvCommand=\

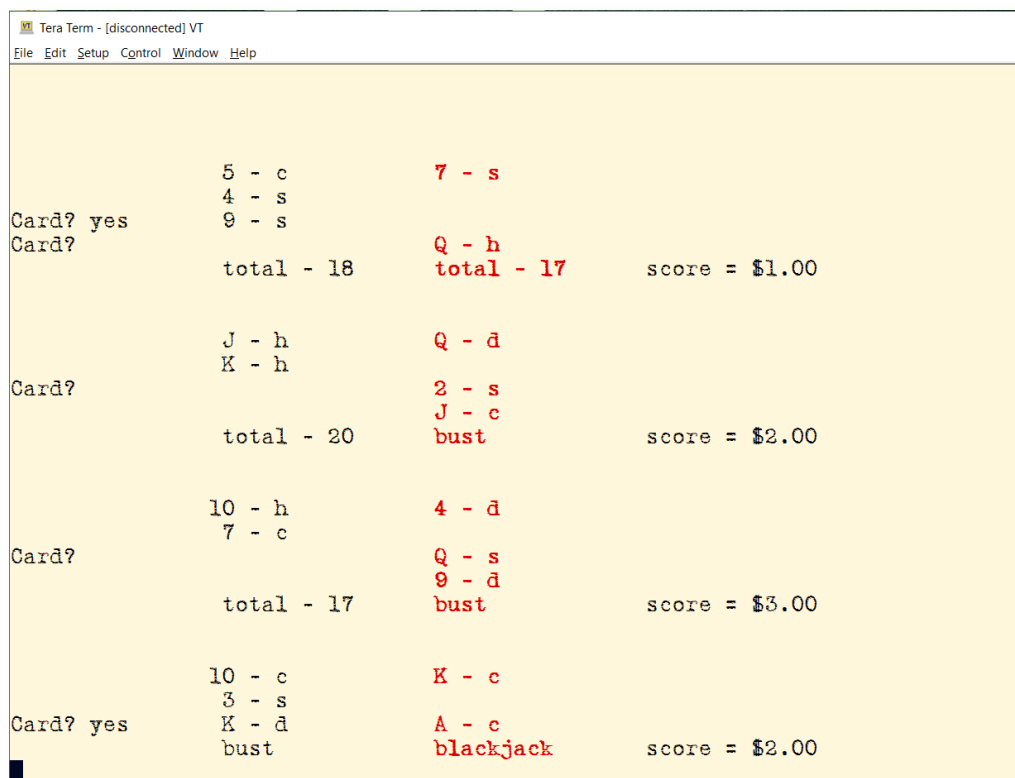
; Black text and yellowish background colors
; for normal characters
VTColor=0,0,0,255,255,229

; Set "bold" typeface to red, to match LGP-30 color shift
EnableBoldAttrColor=on
VTBoldColor=222,0,0,255,255,230
```

The Flexowriter – at least some models – had tabulator stops that were easily adjustable depending on the current application. The complete set of tab stops was arranged on a removable tabulator rack (<http://homepage.cs.uiowa.edu/~jones/flexo/tour.shtml>). While I am not sure whether this was

the case for the Flexowriter model shipped with the LGP-30, it appears that tab settings different from the 8-character increments common today were indeed used. The output from the well-known Blackjack program is correctly aligned for a tabulator spacing of 12 or larger.

The VT-100 terminal has user-adjustable tab stops as well: The “ESC [3g” sequence tells the terminal to clear all pre-defined tab stops; “ESC H” sets a tab stop at the current cursor position. If your terminal program correctly emulates the VT-100, you can prepare text files to set the tabs at various positions. This will allow you to emulate the Flexowriter’s quickly exchangeable tab racks by replaying the desired tab-setting file. Tera Term has a command line option to replay a file upon startup: “ttermpro.exe /R=tab12.tty” will start a terminal with tabs set every 12 characters, if that’s how you have set up the tab12.tty file.



```
Tera Term - [disconnected] VT
File Edit Setup Control Window Help

      5 - c      7 - s
      4 - s
Card? yes      9 - s
Card?          total - 18      7 - s      score = $1.00
                                Q - h
                                total - 17
                                bust

      J - h      Q - d
      K - h
Card?          total - 20      2 - s      score = $2.00
                                J - c
                                bust

     10 - h      4 - d
      7 - c
Card?          Q - s      score = $3.00
              9 - d
              bust

     10 - c      K - c
      3 - s
Card? yes      K - d      A - c      score = $2.00
      bust      blackjack
```

With the right color scheme, font, and tab setting, the terminal does a credible job at conveying the Flexowriter look. (No sounds, unfortunately...)

Drum memory display (HDMI) – optional

Use of the drum memory display is entirely optional. The real LGP-30 did not have such a display, and the replica can be fully used without it. When you really want to follow the operation of the LGP-30 on a bit level, the display may be useful.

Finding a suitable monitor

A monitor with an HDMI input is required. DVI inputs can also be used, with a cheap adapter. (HDMI and DVI video signals are electrically the same, so only a passive adapter plug is required.) The video output has XGA resolution, 1024*768 pixels.

Revision 1 boards (8/2016) have a HDMI output which does not fully meet the HDMI standard: It does not provide a +5V output line, since no +5V supply is available on the board. Apparently many computer graphics cards take the same liberty, so most recent computer monitors should accept the signal without complaints. Many TVs and old monitors however seem to require the +5V line, and are hence not suitable. Board revision 2 (9/2016, with additional power jack on the board) fixes this limitation.

Display modes

The display always shows all 64 tracks of the LGP-30 main memory. Register contents and control signal tracks are also shown in the bottom. Track numbers are indicated on the left and right side of the screen, sector numbers on top. Note that the display shows the sectors in their physical order on the drum; therefore the logical sector numbers follow the 7-step interleave pattern used by the LGP-30.

The display shows a 10-word segment of each track. Low bits are shown on the right. Word boundaries and word structure are delineated by blue bands, with track, sector and opcode fields highlighted in brighter shades.

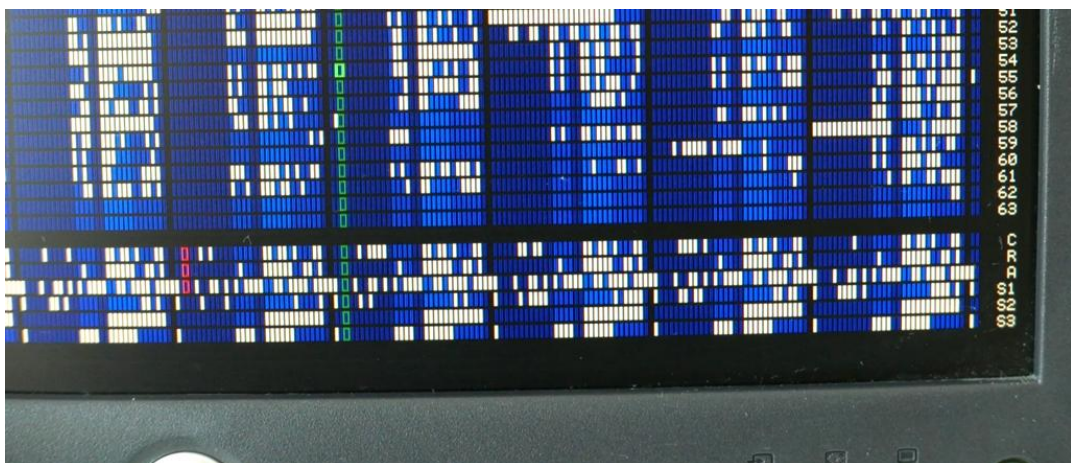
In normal operation, the display is static and can be scrolled left and right with the encoder knob. (Encoder must be in normal mode, with its LED off.)

At slow CPU clock rates (250 Hz and below), the display will switch to scrolling mode. A stationary column of 64 read/write heads is shown in green, and the drum's bit pattern scrolls across it from left to right. The active read head is indicated by a highlighted frame; during write operations this frame switches to red color.

The three dedicated write heads for the re-circulating registers are shown in red, upstream from the read heads. When register contents change, you can observe how the write head refreshes the whole track content with the same repeating pattern, by continuously re-writing what the read head sees 32 bits downstream. The additional read head for the accumulator, used during multiplication, is not shown – the drum does not “know” when the logic unit reads this head anyway.



The drum display shows all 64 tracks as well as the CPU register and sync tracks, but only 10 sectors at a time. Rotate left or right with the encoder knob.



When the logic unit's clock is slowed down to 250 Hz and below, the read/write heads will be displayed, and the drum will move across the heads in real time.

Drum memory storage (XModem) – optional

The magnetic drum of the real LGP-30 retains its contents – main memory and registers – when the LGP-30 is stopped and shut down. After restarting the computer, a program can be resumed at the exact point it had been stopped.

In contrast, this replica stores the drum content in RAM memory; data are lost when the FPGA is powered off. To make the use of complex programs easier – e.g the ACT compiler and libraries, which have to be read from multiple tapes – drum contents can be stored and retrieved via XModem file transfers. To use this feature, you must use a terminal program that supports the classic XModem protocol:

Operation

- The classic XModem protocol is used, with 128 Byte blocks and 1-byte checksums (not CRC).
- Main memory (64*64 words) and register tracks (A, R, C) are stored.
- Download of the drum contents from the FPGA is started via the terminal program. The simulator will recognize the NACK control character sent by the terminal program and will automatically start the transfer. This is part of the standard XModem protocol and should work for all terminal programs that implement XModem.
- Upload of drum contents to the FPGA is also started from the terminal program. It requires sending the reserved “\” character to signal a transfer to the replica, which will then enter reception mode and send a NACK. This function of the “\” character is *not* an XModem standard feature, but a convention adopted for the LGP-30 replica only.

The “\” can either be sent manually from the terminal program; in this case the upload must be started max. 100 seconds afterwards. Alternatively, some terminal programs are configurable to send an XModem upload command automatically. See the “Serial Terminal” section above for an example configuration for the Tera Term program.

Data format

As long as you just store and retrieve drum files, you don’t need to care about the data file format. Here it is anyway. Note that for technical reasons, the file format is *not* compatible with the format used by Christian Corti’s LGP30SIM.EXE!

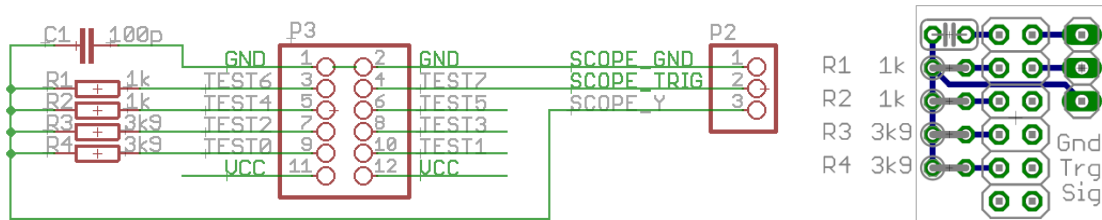
- Data files are 18,432 byte binary files.
- First 16,384 bytes are the main memory data:
Track 0..63 in ascending order; each starting with logical sector 00, followed by sectors in physical order (i.e. logical sectors 00, 57, 50, ...); low byte first.
- Following 2,048 bytes are the register data:
Each stored byte contains one bit each from the A, R, C register tracks, with the following structure: “0 0 0 0 0 A R C” (five padding zeros in the high bits, register data in the low bits). Bytes are stored in the order in which they appear on the physical tracks, starting with drum position 0 (logical sector 0 on the data tracks).

Oscilloscope connection – optional

The LGP-30 used an oscilloscope to display the register contents – accumulator, instruction register, and program counter. This type of display was needed since, due to the bit-serial architecture, the bits only showed up one at a time, at 120 kHz clock rate.

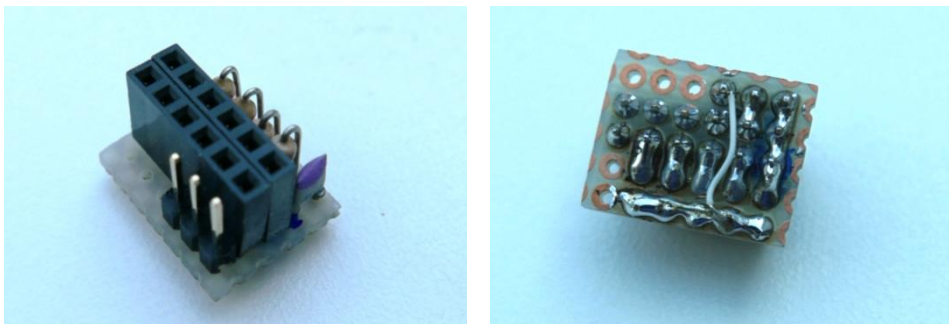
The replica has an LCD to display the same graphical information, so the oscilloscope is not needed. If you want to install one just for kicks, here's how. (The oscilloscope is actually a better display than the LCD, since it follows changes quite a bit faster!)

The test and expansion connector, P3, provides the required signals: A register output, which alternates between bitstreams of the A, R, and C registers, and two offset outputs, which create the vertical offsets. All signals need to be added via a simple resistor network – please see the circuit diagram and layout suggestion below. A trigger signal, which occurs at the beginning of each register word, is also provided.

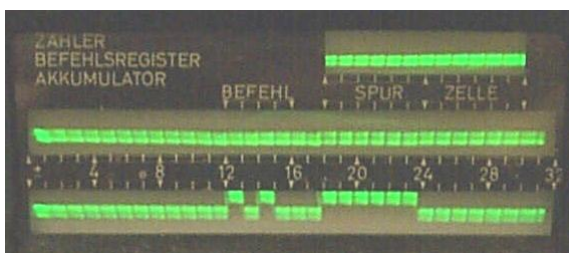


Schematic and layout for an oscilloscope adapter for the expansion port.

Easily hand-wired on perfboard, as shown below.



In the only image of a working LGP-30 display that I know of, the traces have a large vertical width (but are quite crisp in the horizontal direction). I don't know what causes the vertical width and whether the LGP-30 scope was supposed to look like this, but find that it makes the display easier to read. Hence, a “fat trace” component (5 MHz signal) is output on the connector and added to the Y signal via R4 to make the trace appear wider. To reduce the width of the trace, increase the value of R4, or vice versa.



LGP-30 Display.

Image from http://computermuseum.informatik.uni-stuttgart.de/dev/lgp30/lgp30_3.html

The polarity of the provided signals is such that the traces will appear in the same order as in the LGP-30: Counter on top, instruction register in the middle, accumulator below. However, the bit order will be the wrong way round: The LGP-30 drum always outputs the low bit first (needed for ripple carry during addition and subtraction), and its oscilloscope scanned *from right to left* to show the bits in the conventional order, with the low bit on the right. Your oscilloscope is probably fixed to scanning from left to right; and unfortunately the FPGA cannot easily output a ramp signal for use with an X/Y scope.

There is an easy solution to this problem: Turn your scope upside down! ;-)
You will also need to invert the scope's Y input to correct the orientation.



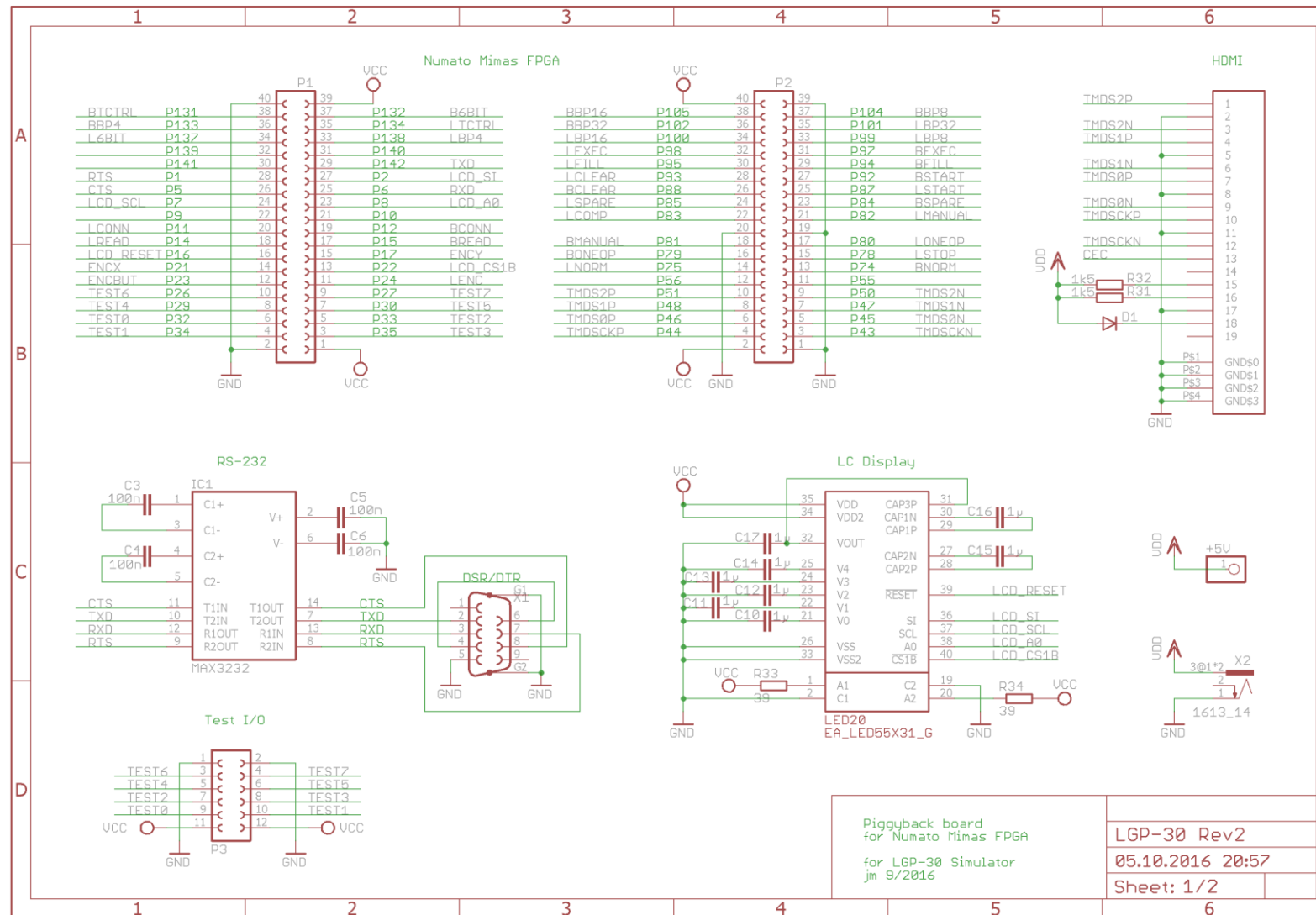
This scope scans from right to left, showing low bits on the right ;-)

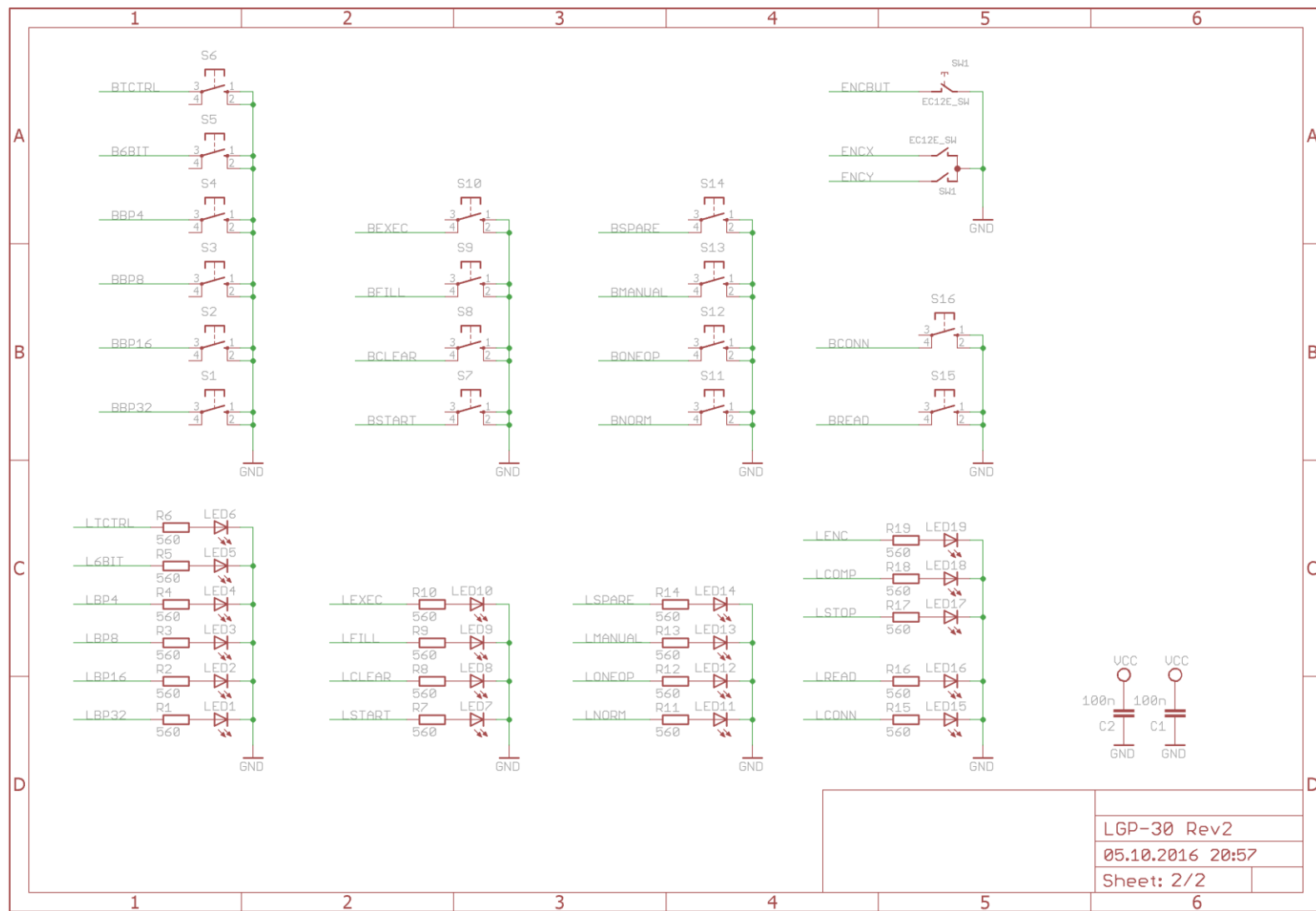
Appendix

Download links

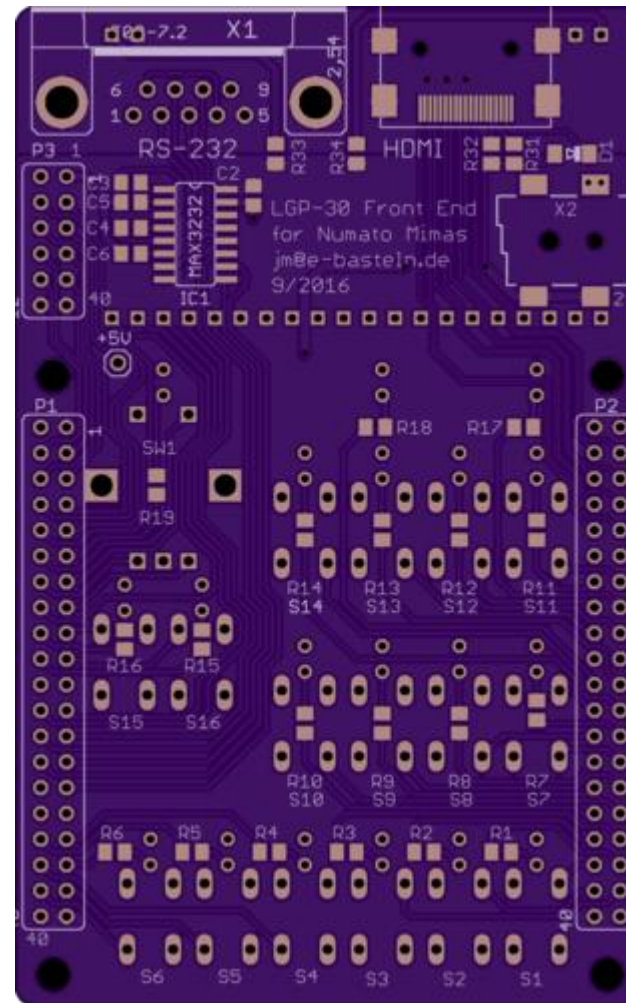
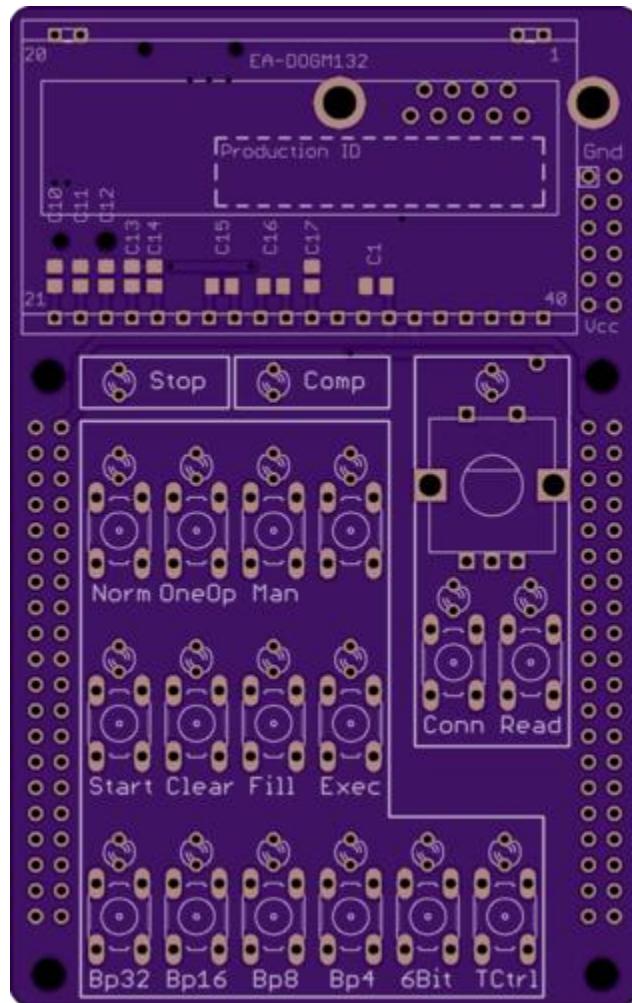
- All my project files (FPGA source code and binary, and the latest version of this document) can be downloaded from www.e-basteln.de/lgp30.
- Tools and documentation for the Numato Mimas Spartan 6 development board are available from <http://numato.com/mimas-spartan-6-fpga-development-board/>.
- For download links to the Xilinx development tools, see the section on FPGA development.
- The University of Stuttgart's computer museum has a great collection of LGP-30 information:
 - Program documentation and ASCII-converted paper tapes:
<ftp://ftp.informatik.uni-stuttgart.de/pub/cm/lgp30/>.
 - Manuals and more program documentation:
<ftp://bitsavers.informatik.uni-stuttgart.de/pdf/royalPrecision/LGP-30/>
 - History and repair of their original, working LGP-30:
http://computermuseum.informatik.uni-stuttgart.de/dev_en/lgp30/lgp30.html.
- The Operation Manual and Programming Manual are specifically recommended as starting points for exploring the LGP-30:
 - Operation Manual ftp://bitsavers.informatik.uni-stuttgart.de/pdf/royalPrecision/LGP-30/LGP-30_Operations_Manual.pdf
 - Programming Manual ftp://bitsavers.informatik.uni-stuttgart.de/pdf/royalPrecision/LGP-30/LGP-30_Programming_Manual_Apr57.pdf

Schematics





PCB layout



Bill of materials

Part numbers refer to www.reichelt.de, and are for reference only. Their online catalog has further details and data sheets if any designation is ambiguous. (Catalog can be switched to English.)

| Schematic Ref | Quant | Item | Reichelt No. | Comment |
|----------------------------------|-------|---------------------------------------|----------------------------------|---|
| C1 - C6 | 6 | Capacitor 100nF, SMD 0805 | X7R-G0805 100N | multilayer ceramic capacitor |
| C10 - C17 | 8 | Capacitor 1µF, SMD 0805 | X7R-G0805 1,0/25 | multilayer ceramic capacitor |
| R1 - R19 | 19 | Resistor 560 Ohm, SMD 0805 | SMD-0805 560 | series resistors for 2mA LEDs |
| R30 | 1 | (leave unpopulated) | | PCB rev 1: unpopulated, rev 2: removed |
| R31, R32 | 2 | Resistor 1k5, SMD 0805 | SMD-0805 1,50K | I ² C termination HDMI |
| R33, R34 | 2 | Resistor 39 Ohm, SMD 0805 | SMD-0805 39,0 | series resistors LCD backlight |
| LED 1 - 6, LED 11-16 | 12 | LEDs green, diffused, 3mm | LED 3MM 2MA GN | 2mA low current |
| LED 7 - 10 | 4 | LEDs yellow, diffused, 3mm | LED 3MM 2MA GE | 2mA low current |
| LED 17 - 19 | 3 | LEDs red, diffused, 3mm | LED 3MM 2MA RT | 2mA low current |
| U1 | 1 | LCD, Electronic Assembly DOGM132-5 | EA DOGM132S-5 | transmissive, inverse, FSTN |
| LED20 | 1 | LED backlight for EA DOGM | EA LED55X31-G | green |
| IC1 | 1 | MAX3232CSE | MAX 3232 CSE | RS-232 level converter, 3V power |
| S1 - S16 | 16 | Push-button, short stroke, 6x6mm | TASTER 3301 | S16 is optional (spare) |
| SW1 | 1 | Rotary encoder with push-button | STEC12E08 | ALPS EC12E24244 |
| (none) | 1 | Knob for 6mm flattened axis | KNOPF 10-150E | for encoder, optional |
| K1 | 1 | HDMI connector | HDMI SMD PEB | with SMD mounting tabs |
| P1, P2 | 1 | Pin header 2*40, 2.54mm pitch | SL 2X40G 2,54 | split into 2 headers, 2*20 each |
| P3 | 1 | Pin header 2*6, 2.54mm pitch | SL 2X10G 2,54 | optional. Part is 2*10, shorten to 2*6 |
| (none) | 1 | precision pin socket 1*32 | SPL 32 | for LCD, split into 20+2+2 pins |
| X1 | 1 | D-Sub jack, 9 pins, angled, 7.2mm | D-SUB BU 09US | short US type! 7.2mm front edge > pins |
| <i>For board revision 2 only</i> | | | | |
| D1 | 1 | Schottky diode BAT46, Mini Melf | BAT 46 SMD | UF = 0,25V, 150 mA |
| X2 | 1 | DC power jack 2mm, SMD | LUM 1613-14 | optional, Lumberg 1613-4 |
| (none) | 1 | DC power plug, angled, 2.1mm | HSW 21-9 | optional, matches jack X2 |
| (none) | 1 | Pin header, 1 pin | MPE 087-1-002 | optional, mates with Mimas +5V socket. Use 1 pin only. |
| (none) | 1 | Pin socket, 2 pins (Mimas +5V) | BL 1X05G7 2,54 | optional, populate on Mimas board! use 2 pins only for +5V and GND |