Pulse programming for Dummies

Börge Hemmerling

Institut für Experimentalphysik
University of Innsbruck
boerge.hemmerling uibk.ac.at
This manual is work in progress

Contents

1 Introduction

This document is meant to be an introduction to the pulse-programmer. It describes the basic functionality and the guides a new user to the setup process of the pulse programmer.

More in depth documentation may be found on the pulse programmer homepage: `http://pulse-programmer.org`

This manual is work in progress. If you find any errors or if you have any additions write an email to the author.

2 What can the box do?

In a typical quantum computing setting, the user would like to perform operations on a qubit. These operations can be reduced to a sequence of pulsed electromagnetic waves with certain amplitudes, phases, carrier frequencies, and time durations. A recurring problem in these experiments is the transfer of desired pulse sequences from the user to the qubit using minimal resources and introducing as few errors as possible.

There are two parts to solving this problem: designing a language for specifying arbitrary pulse sequences using *pulse programs* and designing a device, called a *pulse sequencer*, for translating this language efficiently and accurately into digital outputs. This project contains a specialized Pulse Control Processor (PCP) for outputting digitalpulses and a corresponding assembly language (PCP assembly); it also contains user interfaces and development tools to make using the pulse sequencer and writing pulse programs easier and more intuitive.

By itself, the sequencer only controls the timing of digital outputs (bits) and is agnostic to how these outputs are used. These bits are interpreted by a *waveform synthesizer*, which combines them with a carrier wave to produce modulated analog output. Each analog signal, known as a *channel*, feeds into an apparatus which is specific to the chosen qubit; multiple bits from the sequencer can be assigned to a single channel, and one sequencer can control multiple channels at once. This project contains also a waveform synthesizer which is based on the concept of direct digital synthesis (DDS).

The hardware to this project consists of 3 major blocks:

- sequencer main board: Contains the Pulse Control Processor (PCP)
- breakout board: Provides an interface between the (PCP) and the other periphals
- RF synthesizer board: Generates exactly timed radio frequency pulses.

2.1 Features

The main features of a typical system are:

- Pulse program flow control (Triggers, finite loops)
- Up to 32 exactly timed digital outputs
- Up to 16 exactly timed radio frequency outputs
- 8 trigger inputs for program flow control
- Phase coherent switching of radio frequency pulses

2.2 Figures of merit

This section refers to a system consisting of a single PCP baord a breakout baord and an AD9910 synthesizer board.

Available TTL output channels	32
Available TTL trigger inputs	8
Max Nr of DDS boards	16
Minimum time step	10ns
Maximum wainting time	2s

The figures of merit for the RF synthesizer are:

Maximum output frequency	$\approx 350\,MHz$
Minimum output frequency	$< 10\,MHz$
Number of coherent frequencies	16
Frequency switching time	$\leq 150\,ns$
Phase offset accuracy	$2\pi\ 2.4 \cdot 10^{-4}$
Signal to noise ratio	$50\,dBc$
Frequency resolution	$0.18\,Hz$
Maximum output level	$\approx -1\,dBm$

3 What do you need?

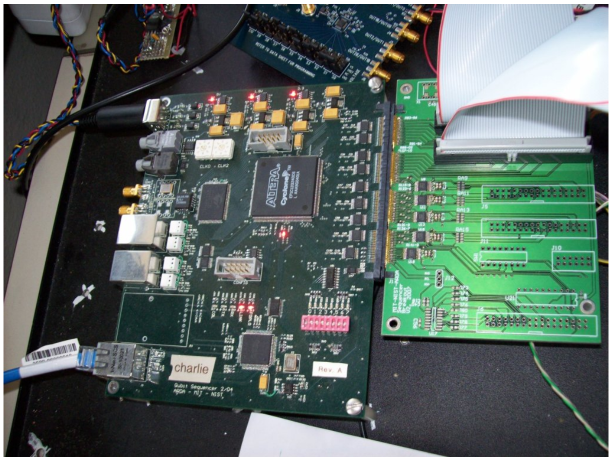
This section gives an overview over the electronic parts and software which is needed to operate a pulse programmer.

3.1 Hardware

3.1.1 Sequencer board

The sequencer board is the main board of the pulse programmer. It contains an altera Cyclone I FPGA. It provides following interfaces:

- Ethernet: For communication with the experiment control computer.
- PTP: For communication with another sequencer main board (not used)
- LVDS: Bus system with digital outputs and inputs controlling the other parts of the pulse programmer and direct digital in- and outputs.
- Clock: The clock of the main board (100MHz)



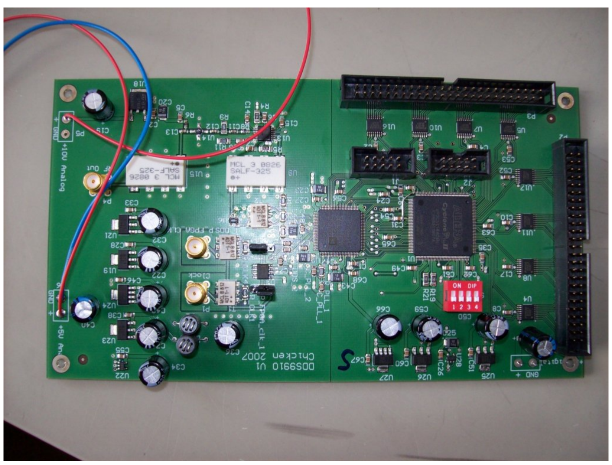
The image shows the main board with an unfinished breakout board attached.

3.1.2 Breakout Board

The breakout board converts some of the LVDS outputs of the main board to TTL outputs which are used as digital in- and outputs.

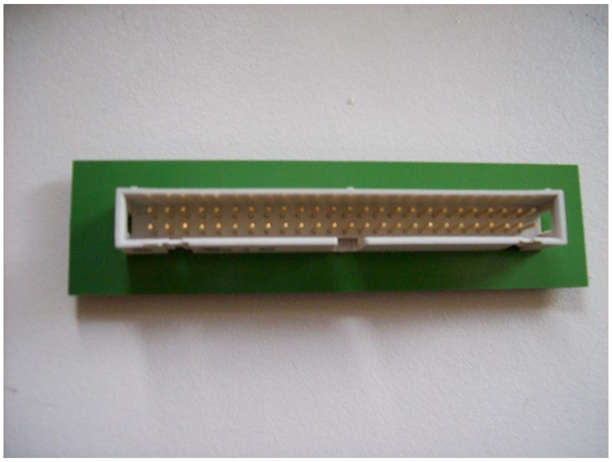
It provides connectors for the bus system controlling the DDS board

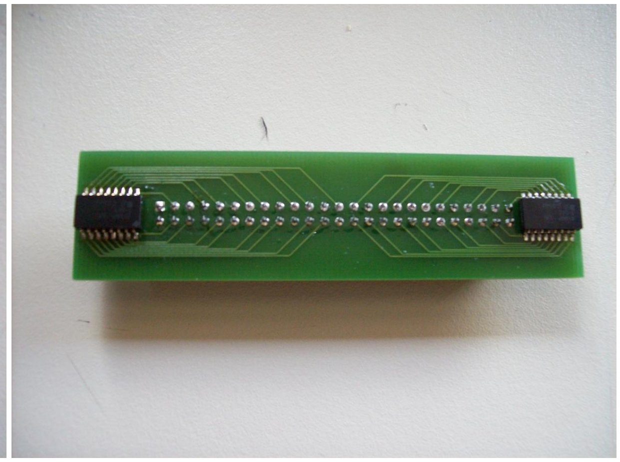
3.1.3 DDS Board



The DDS board is based on an analog devices AD9910 direct digital synthesizer and an altera Cyclone II FPGA. It is connected to the breakout board with two 50 pin flat ribbon cables.

3.1.4 100 Ω Terminator Board





As the bus system controlling the DDS board is based on the LVDS standard it needs a termination resistor array. For each of the two flat ribbon cables a terminator board is needed.

3.1.5 USB Blaster

To install the firmware on the boards you need a JTAG programming cable which is available e.g. here `http://www.absolute-data-services.co.uk/terasicblaster.htm`. The FPGAs can be programmed with two different connectors / methods:

- JTAG: Volatile programming method. Is used for debugging in combination with the on-chip logic analyzator Signal-Tap II. (See altera documentation for more details)
- ASP (Active serial programming) USed for permanent programming of stable firmware.

3.1.6 Clock

Clock Divider Board
Marconi
on board quarz (dip switch?)

3.1.7 Miscellaneous

Order list for Farnell and RS

item	part number	pieces per box
Flachkabel 50-polig AWG28 1,27mm	RS 105-5124	<2m
Kabelbuchse IDC 2,54mm 50-polig	RS 323-7952	$4 + 2\times\sharp$ of DDS boards
Zugentlastung fr IDC-Buchse 50-polig	RS 192-7625	$4 + 2\times\sharp$ of DDS boards
AMPHENOL SPECTRA-STRIP - 132-2802-234 - RIBBON CABLE, 34WAY, PER M	Farnell 1207481	<2m
Kabelbuchse IDC 2,54mm 34-polig	RS 192-7489	$4 + 2\times\sharp$ of DDS boards
Zugentlastung fr IDC-Buchse 34-polig	RS 192-7596	$4 + 2\times\sharp$ of DDS boards
Widerstandsnetzwerk 8x100R	RS 522-4942	2 (for breakout board)
Mini 2.5mm Stecker verriegelt	RS 329-9944	
Crimpbuchse AWG 22-30 KK 4809	RS 467-598	1 package
Buchsengehuse KK 6471 2,54mm 12-polig	RS 296-5038	3
KK 4030 Stiftleiste 12-polig	RS 479-226	3
Stiftleiste KK5,08 3-polig	RS 230-5363	10
Buchsengehuse KK5,08 3-polig	RS 230-5161	10
Steckverbinder,Klemmleiste,Lsterklemme, Schraubklemme,PA,natur,400V,4mm,12-polig	492-8716	1
SMA male-male ADAPTER	Farnell 1169638	0
SMA female-female ADAPTER	Farnell 1169639	0
SMA male-female 90 ellbow	Farnell 1169636	evtl. $3\times\sharp$ of DDS boards
SMA housing connector	Farnell 1169637	8
BNC housing connector	Farnell 1205963	40
SMA cable RG174 2XSMA length 0.25m 0-0	Farnell 1056170	$5 + 3\times\sharp$ of DDS boards
SMA cable RG174 2XSMA length 0.25m 0-90	Farnell 1056176	0
SMA cable RG174 2XSMA length 0.25m 90-90	Farnell 1056195	0
BNC end-resistor 50-OHM	Farnell 1170188	8
Network PATCH KABEL 0.3m BEIGE 0.3M	Farnell 1526126	1
Network housing connector shielded	Farnell 1122292	1
Jumper gold	Farnell 1097979	$5+ 3\times\sharp$ of DDS boards
LED green 3mm with series resistor	Farnell 1003391	35
LED clip for 3mm LEDs	Farnell 1208877	35
Bananenbuchse black	Farnell 1176428	1
Bananenbuchse red	Farnell 1176427	1
Mechanical switch	Farnell 9473378	3
Laborkarte (Ltkarte)	Farnell 1172173	1
Nylon Spacer for circuit boards M3 X 31.5mm	Farnell 963252	$8 + 4\times\sharp$ of DDS boards
19ZOLL housing 3HE 350MM UNBELUEFTET	Farnell 1277472	1
or: 19ZOLL housing 3HE 350MM BELUEFTET	Farnell 1277466	1
backwall for housing 3HE 19ZOLL	Farnell 1277475	1
Power connector for main boardd	Farnell 152209	1

Order list for Minicircuits for configuration with 6 DDS boards

item	part number	pieces per clock
Power Splitter	ZCSC-8-13+	2
Low Pass	VLF-190+	1
50 Termination	Anne-50+	7
Amplifier	ZHL-2010+	1
Attenuator	VAT-6+	1
Amplifier (also needed for amplifying DDS output of max -14 to -13 dbm for AOM use)	ZFL-1000	(1) optional
Attenuator	VAT-4+	(1) optional

3.2 Software

In this section the software is given that you need to install to communicate with the box. All the software is avaible for Linux and Windows sytems.

3.2.1 Python

Download the latest python interpreter from `http://python.org/`and install it. Furthermore, you need to install Ipython for running the hardware tests later on. You can get that from `http://ipython.scipy.org`.

3.2.2 Hg Mercurial

For the python server we use Hg Mercurial as a CVS system. Download it from `http://selenic.com/mercurial`and install it. If you use the windows operation system you might want to use TortoiseHG

3.2.3 Python Server

The python server is available as a repository in the internet. Having Mercurial installed, in order to download the server execute the command

hg clone static-http://hg.brainity.com

or use the user interface in Windows.

3.2.4 Cygwin

Cygwin is only needed if you want to modifiy the sequencer main board firmware. If you don't know what this is you probably don't need to install Cygwin.

Install the latest Cygwin version from `http://cygwin.org/`and make sure you installed the packages m4 and make.

3.2.5 USB Blaster Driver

In order to use the USB blaster cable to communicate with the programming ports of the boards you need to install its driver. It's available at `http://www.altera.com/support/software/drivers`.

3.2.6 Altera Quartus II

You will need Quartus II if you want to modify or program the firmware of the any FPGA. In order to install the firmware of the boards you need to install Quartus II from `http://www.altera.com/support/software/sof-quartus.html`.

3.2.7 Wireshark

Install Wireshark from this location `http://www.wireshark.org/`. This tool is needed to check wether the box communicates in the beginning with your computer.

3.2.8 Firmware of the sequencer main board

Get the firmware from the pulse programmer download site `http://pulse-programmer.org` Building the firmware in a Cygwin environment with M4 installed:

- make sequencer_top.vhd
- make sequencer_top.map.eqn
- Open the project sequencer top.qpf in Quartus. Press Ctrl+L

3.2.9 Firmware of the DDS Board

The firmware of the DDS board is available at the SourceForge download page at:
`https://sourceforge.net/project/showfiles.php?group_id=129764&package_id=265102`

3.3 Documentation

Check the Documentation Wiki on the pulse programmer homepage for an updated list of manuals, tutorials and guides.
`http://pulse-programmer.org`

3.4 Schematics

The schematics of the sequencer main board and the breakout board may be downloaded at the pulse programmer download site.

The schematics of the DDS baords are not available online yet. A copy may be requested from Philipp Schindler (philipp.schindler@uibk.ac.at)

3.5 Data Sheets

The data sheets of the used chips can be found at the Analog Devices wegpage. Here are the direct links

FPGA Cyclone I	`http://www.altera.com/literature/lit-cyc.jsp`
FPGA Cyclone II	`http://www.altera.com/literature/lit-cyc2.jsp`
AD9910 (DDS)	`http://www.analog.com/UploadedFiles/Data_Sheets/AD9910.pdf`
AD9513 (Clock Divider up to 800 MHz)	`http://www.analog.com/UploadedFiles/Data_Sheets/AD9513.pdf`
AD9514 (Clock Divider up to 1.6 GHz)	`http://www.analog.com/UploadedFiles/Data_Sheets/AD9514.pdf`

4 Setting up the Hardware

This section tells you how to assemble the hardware.

4.1 Power Supply

You need the following voltage supplies for the components

Component	Voltage [V]	Current [A]
PCP Board	6	~ 1
Breakout Board	?	?
Clock Divider AD9513	3.3	?
DDS Board	5	0.3 (per board)
	10	<0.1 (per board)

This can be done via some voltage regulator on a separate board (see above) or separate power supplies. For making things easier in the future a new power supply board is currently under construction (see ???).

4.2 The Clock Distribution System

The clock of the FPGA on the PCP board needs 100 MHz reference clock, whereas each DDS board needs both 100 MHz (for its own FPGA) and 800 MHz reference for its DDS chip. The clock of the PCP board can either be provided by an on-board quarz or externally which is choosen via the dip switch close to the sma connectors on the board. In our case we use a Marconi 2024 to provide 800 MHz and the frequency divider board AD9513 from Analog to divide it down to 100 MHz. The complete clock distribution system for the PCP and six DDS boards is shown in Fig. 1.

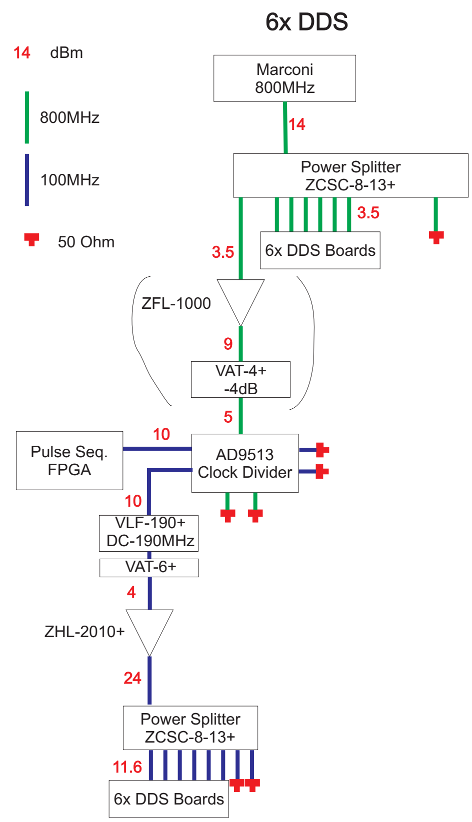


Figure 1: Shown is the clock distribution system we used for one PCP board and six DDS boards. The label of each component corresponds to the Mini-Circuits part numbers.

4.3 sequencer main board

4.3.1 Power Supply

The sequencer main board needs a voltage 6V and draws a current of about 1A.
Add pinout of power supply connector

4.3.2 Clock setup

The sequencer main board has an internal 100MHz oscillator and two SMA connectors for an external clock input. With the switch J?? the clock source can be determined.

4.3.3 IP Address of the main board

You need to set the IP address of the main board. This needs to be done via the dip switch on the pcp board. The pinout is:

reset	ip1	ip2	ip3	ip4	dhcp
0	1	0	1	0	0 (example)

The dhcp port should be set to 0 (OFF). The IP address is given by

192.168.0.X where X = 220 + ip1 + 2 * ip2 + 4 * ip3 + 8 * ip4

In the example given above the IP address would be 192.168.0.225.

4.3.4 Programming the firmware

The firmware is programmed via the active serial programming (ASP) port which is labeled "config" Download the Sequencer main board firmware from the pulse-programmer.org download site. The download is called sequencer-firmware. Only the binary file is needed: e.g sequencer-firmware-0.29.tar.bz2

4.4 Breakout Board

Benches

TTL cable and bnc ports

4.5 DDS Boards

4.5.1 Programming the firmware

The firmware is programmed via the active serial programming (ASP) port which is labeled J1.

4.5.2 Connecting the DDS board

The connectors P1 and P2 of the DDS board should be connected to the breakout board as follows:

- P100 → P3
- P101 → P2

4.5.3 Terminating the LVDS bus

Each of the two cables which are used for the LVDS bus have to be terminated seperately with one terminator board as shown above.

4.5.4 Connecting the clocks

In the standard configuration the DDS board needs a 100MHz clock for the FPGA and a 800MHz clock for the DDS. It is prepared to be used with a single 100MHz clock as well. One can use the internal PLL of the DDS to scale up the clock. By doing so the signal to noise ratio of the DDS output is significantly lowered.

The clock distibution is set by the two jumpers P_FPGA_clk_1 and P_FPGA_clk_2.

The clocks have to be connected in the following:

- T_DDS_FPGA → 100MHz
- P1 → 800MHz

5 Testing the System

5.1 sequencer main board

Install and configure the python server. The instructions for configuring the software are available in the `README` file in the root director of the python server. Try one of the hardware test scripts shipped with the server. Make sure that the option NONET is set to False! If the hardware test resumes without a 'No Pulse Protocol reply received' error, the main board is configured correctly

5.2 Breakout Board

Try the hardware test script shipped with the python server and test the TTL outputs of the breakout board with an oscilloscope. If a channel does not behave as expected check the LVDS to TTL converter chips and the connector to the main board.

5.3 DDS Board

The testing routines for the DDS baord and the LVDS bus system are available at:
`http://pulse-sequencer.sourceforge.net/innsbruck/AD9910/`

5.3.1 lvds bus

lvds bus dds output phase coherent switching

6 Specifications

6.0.2 For Loops

one for loop

show debug sequence

TTL Pulses	Time Difference (total)	Time Difference per Pulse
100	10.92 μs	109.2 ns
200	21.9 μs	109.5 ns

corresponds to the amount of nops

7 Troubleshooting and Unresolved Bugs

7.1 No pulse transfer protocol reply received

If you get the following error message

No pulse transfer protocol reply received

try to switch off the box, switch it on again and flush the arp cache of your computer. In a command line type

arp -d

That should solve the problem.

7.2 TTL Output is only 3.3 Volts

At the moment all TTL channels - if set to high - give out 3.3 Volts instead of standard TTL 5 Volts. This can cause severe problems if a 'real' TTL is needed. This problem is currently tackled by a new breakout board for the TTL channels which includes a buffer chip to provide the needed 5 Volts.

7.3 DHCP

7.4 Ground on the FPGA?!?

8 User Manual for the Python Server

For a just slightly outdated reference of programming the pulse-programmer with the python server check the documentation available at
`http://sequencer.brainity.com`

8.1 How to configure the Python server

8.2 TTL Pulses

what happens to inverted channels

8.3 RF On

8.4 RF Pulses

8.5 For Loops

8.6 Subroutines

8.7 The Digital Ramp Generator

Flux Diagram of the Python Server