This document is for:

- PCB PT8620 rev. 0 prototype with firmware LTC_v.1.5.hex
- Mainframe firmware pt5300_v6.6.bin (checksum 8851)

Longitudinal Time Code module – PT8620 (rev. 0 prototype)

This optional board consists mainly of the following parts:

- A PSoC microprocessor (µP).
- A VITC decoder (Vertical Interval Time Code).
- An LTC generator (Longitudinal Time Code).
- An RTC (Real Time Clock).

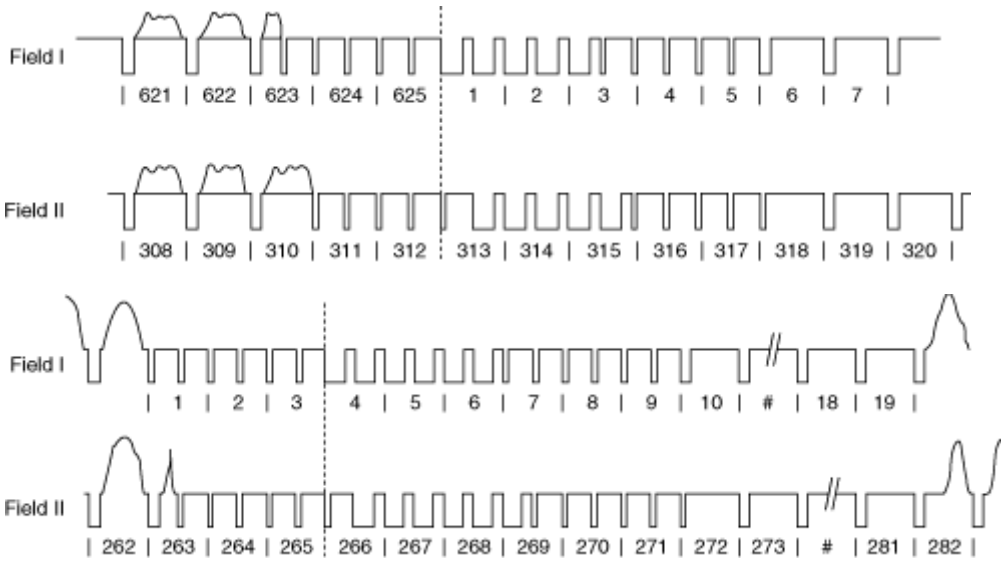
General Information

The purpose of the PT 8620 is to generate an LTC data stream as a balanced audio signal from a number of time sources:

- The on-board battery backed up real time clock (RTC).
- The on-board VITC decoder.
- GPS time supplied from the mainboard.

Clocking considerations

The mainboard signals FF_G (PAL) and FF_M (NTSC) are frame start pulses. FF_G is a positive 64µs wide pulse at horizontal line #1 and FF_M is a positive 64µs pulse at horizontal line #4. These correspond to vertical sync positions in the video signal.



Unfortunately, the frame sync pulses are not stable enough to be used for generation of the LTC telegram clock as originally intended by using the partly analogue PLL. It has been seen that the frame sync pulse can jump in position causing corrupt LTC data. LTC clocking is now based on the OCXO 27MHz from the mainframe.

Schematic description of hardware

The microprocessor (µP) fetches the time from the on-board sources via a 3.3V I²C interface (RTC address 0x68), (VITC decoder address 0x5D).

The GPS time is fetched via the mainboard. The communication with the mainboard is handled with a separate 5V I²C interface with the PT8620 slave address 0x55.

The partly analogue PLL circuit is generating approx. 4 kHz pulses locked to the delayed frame sync pulse, FFD, produced by the µP. The delay is based on LTC specifications (64 µs). The 4 kHz signal was formerly used for clocking out the LTC data serial shift register but is now replaced by a digital on chip solution.

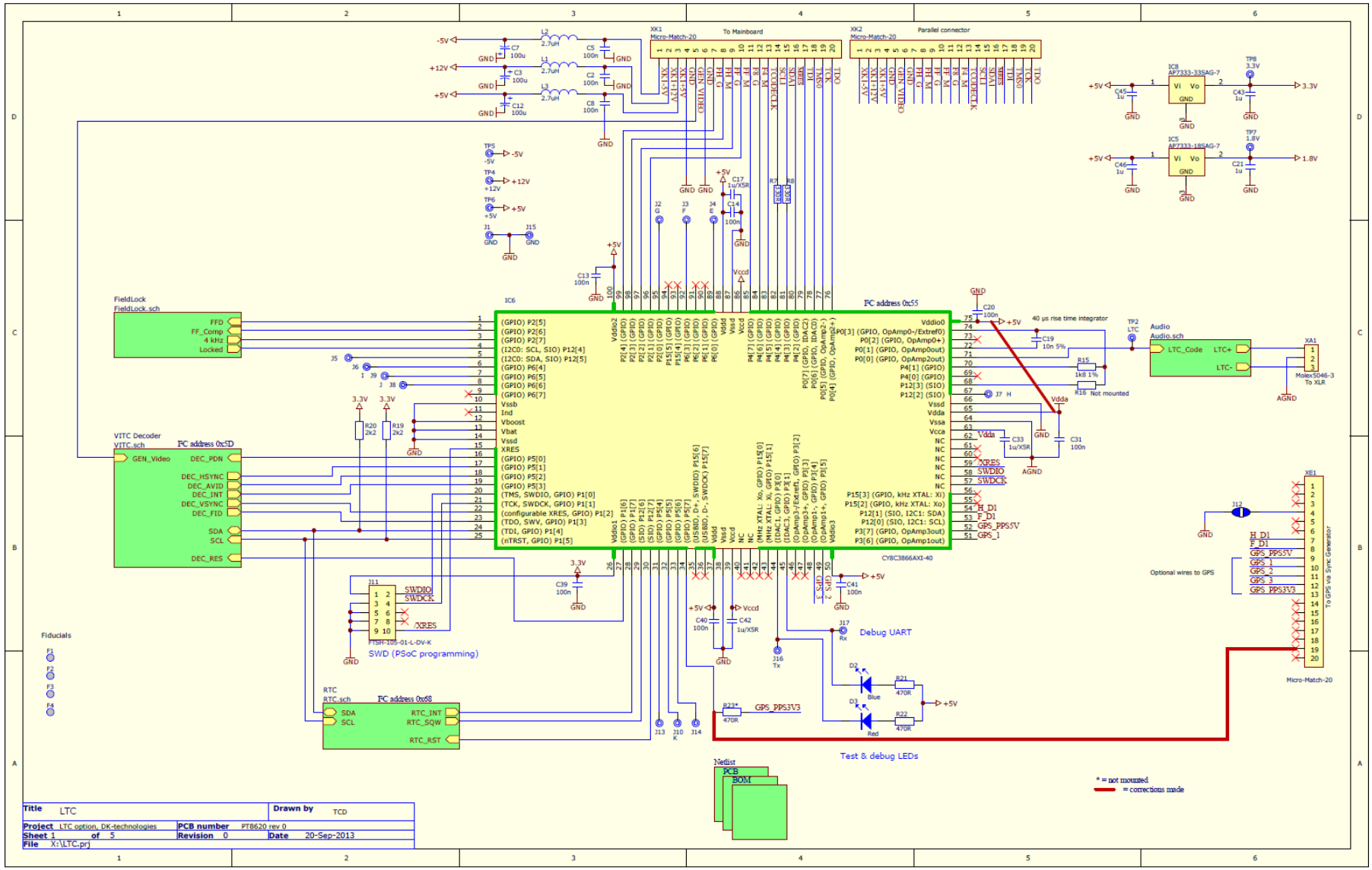
The VITC decoder is a stand alone chip and VITC data is fetched directly from here.

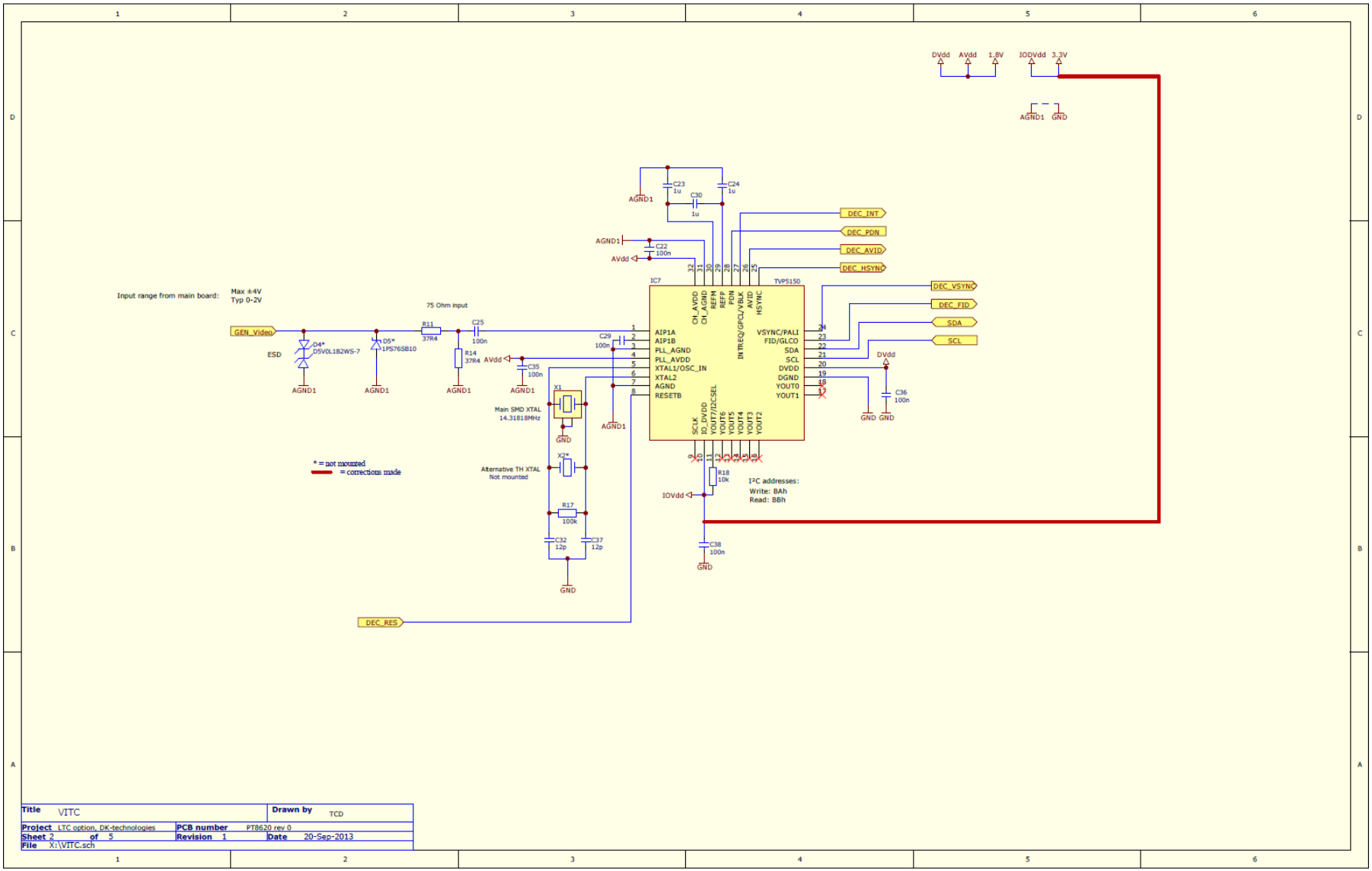
The RTC is a 10 ppm drift, internal crystal controlled, battery backed up second counter. The counter's registers will be converted into a valid time format by the µP.

The LTC digital data is guided to a µP on-chip integrator to provide the correct rise/fall time (40 µs), then to a voltage divider, a low pass filter and a balanced audio op-amp to provide an LTC signal according to LTC specifications. Expect outputs around 1.8 Volts peak-to-peak on each balanced channel. Note the balanced LTC out connector has GND in the centre, pin 2, while an XLR socket typically has pin 1 connected to GND.

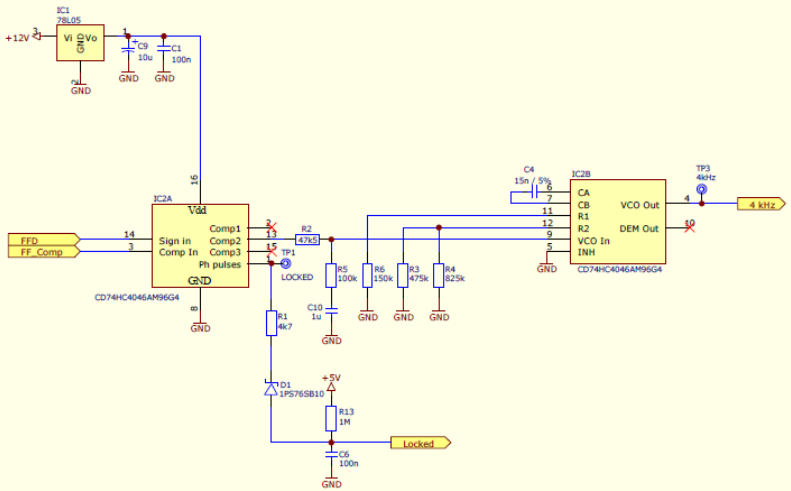
The connectors XK1 should be connected to the mainboard's XK1. XK2 is a parallel to XK1 and can be used for general purpose. XE1 should be connected to the GPS board to look for valid pulse per second (PPS) pulses.

Note the three bold red wires in the schematic are changes to this prototype board that should be made manually on the printed circuit board with thin wires after the production.

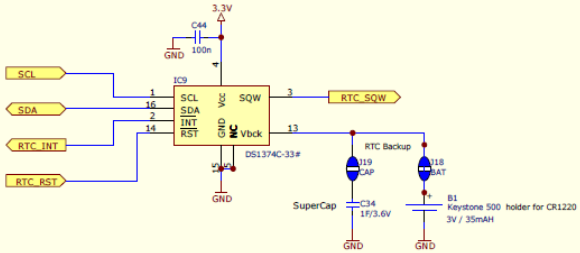

XK1
Micro-Match-20
To Mainboard
XK2
Micro-Match-20
Parallel connector
IC6
CY8C3866AXI-40
I²C address 0x55
FieldLock
FieldLock.sch
VITC Decoder
VITC.sch
I²C address 0x5D
RTC
RTC.sch
I²C address 0x68
SWD (PSoC programming)
Debug UART
Test & debug LEDs
Audio
Audio.sch
40 µs rise time integrator
Optional wires to GPS
To GPS via Sync Generator
Fiducials
Netlist
PCB
BOM
* = not mounted
= corrections made
Title LTC
Drawn by TCD
Project LTC option, DK-technologies
PCB number PT8620 rev 0
Sheet 1 of 5
Revision 0
Date 20-Sep-2013
File X:\LTC.prj



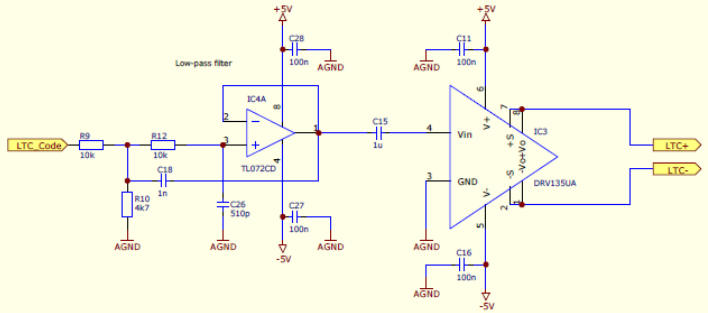
Title VITC		**Drawn by** TCD	
Project LTC option, DK-technologies	**PCB number** PT8620 rev 0		
Sheet 2 **of** 5	**Revision** 1	**Date** 20-Sep-2013	
File X:\VITC.sch			

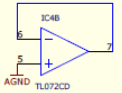


Title	FieldLock	**Drawn by**	TCD
Project	LTC option, DK-technologies	**PCB number**	PT8620 rev 0
Sheet	3 of 5	**Revision**	0
File	X:\FieldLock.sch	**Date**	20-Sep-2013



Title RTC			**Drawn by** TCD
Project LTC option, DK-technologies	**PCB number** PT8620 rev 0		
Sheet 4 **of** 5	**Revision** 0	**Date** 20-Sep-2013	
File X:\RTC.sch			





Title Audio		**Drawn by** TCD	
Project LTC option, DK-technologies	**PCB number** PT8620 rev 0		
Sheet 5 **of** 5	**Revision** 0	**Date** 20-Sep-2013	
File X:\Audio.sch			

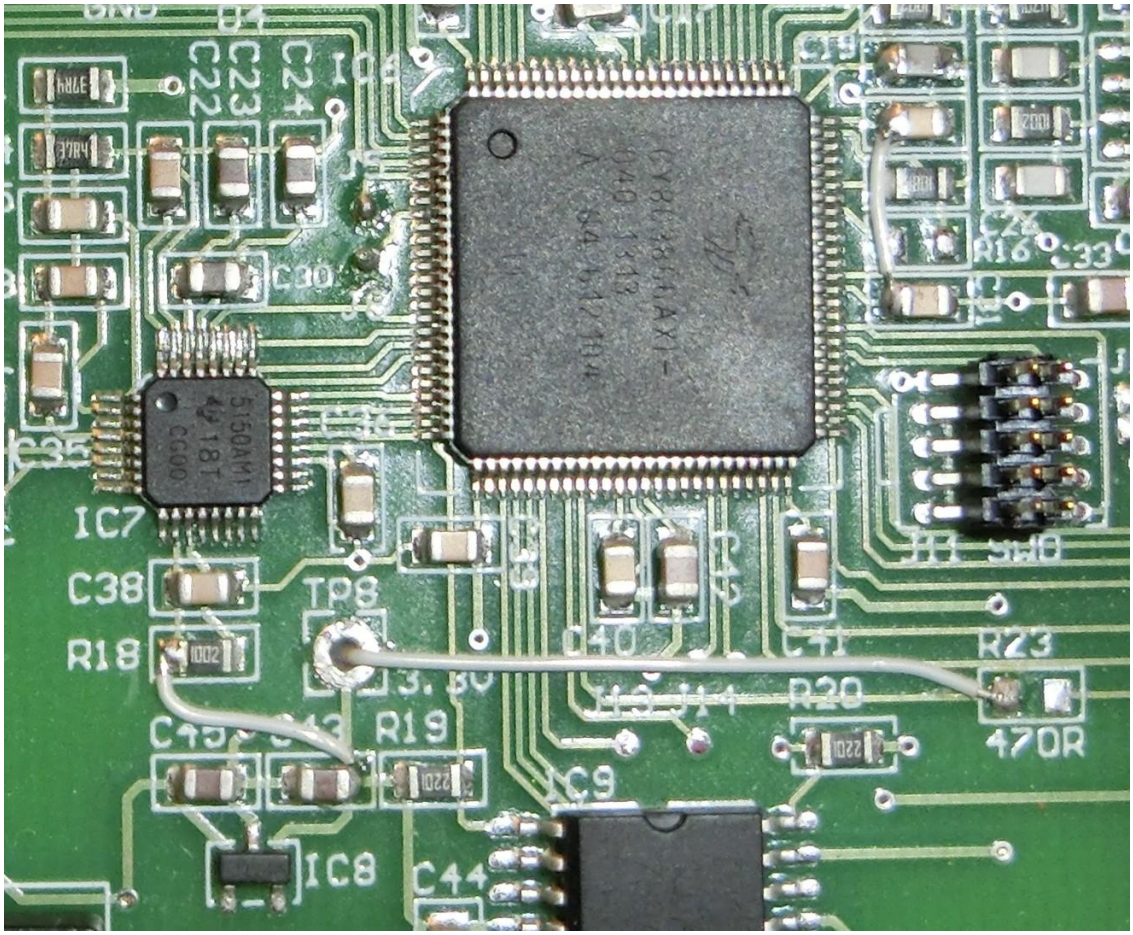
Manually mounted wires

This board is a prototype. Three additional connections must be made manually after PCB production to correct layout faults and interface connections updated after the design phase. The following pictures show how they could be soldered.

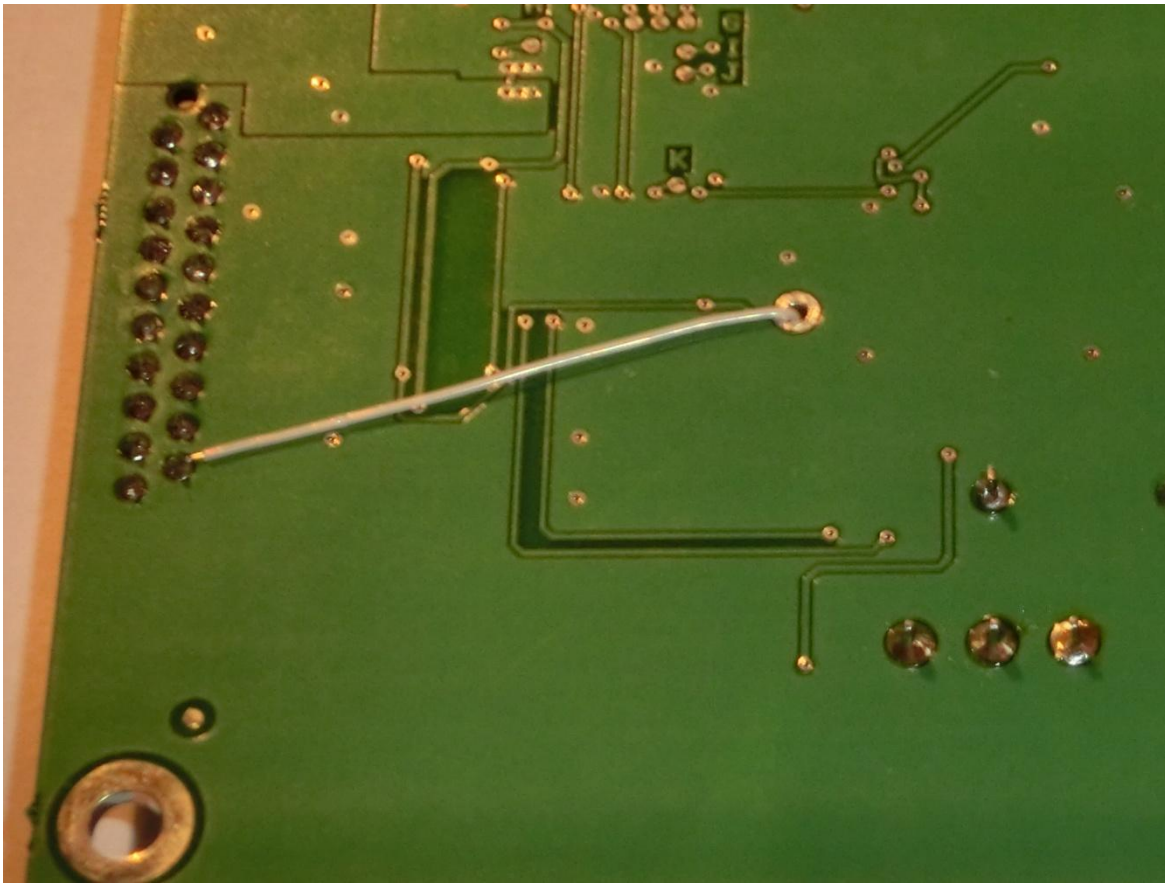
Connect left side of R18 to right side of C43 to provide 3.3V to the video decoder chip IC7.

Connect left side of C20 (symbol not shown) to left side of C31 to provide 5V to analogue section of IC6.

Remove R23 (if mounted) and connect left side of R23 through the TP8 hole to pin 19 of connector XE1 on the bottom side to provide 3.3V GPS PPS signals to the µP.



Wires mounted manually after production, top



Wires mounted manually after production, bottom

Schematic description of PSoC design

This chapter explains the functionality of the components imported into the PSoC. The file LTC_PSoC_datasheet.pdf contains the schematics and other information about the flashed hardware design inside the CY8C38AXI PSoC.

Schematic sheet: Frames

The VideoSystemSelector toggles between PAL and NTSC. The FF_G or FF_M signals are fed to the PWM_2 component that delays the signals 64µs according to LTC specifications. This delayed signal (DelayedFrame) indicates the start of the LTC telegram. The delayed signal is also fed into the external PLL circuit as 'signal in'.

Schematic sheet: PLL

This 'PLL' is a PWM generating a 2x clock rate for the LTC generator. The PLL function is implemented by adjusting the PWM period by ±1 to tune the LTC telegram into the same position as the DelayedFrame.

PMW settings for PAL: Input 42 MHz, period 10499 gives 4000 Hz
PWM settings for NTSC: Input 42 MHz, period 8758 gives 4795 Hz
The PWM duty cycle is not 50% which is OK here.

Schematic sheet: LTC digital

LTC generator:

Control_Reg_1 resets SPI and enables audio output. SPIM_1 sends out LTC serial data with a rate of LTC_Clock/2. isr_SPIM_1 is trigged if SPIM_1 runs out of data.

Bi-phase modulator:

Discrete logic explanation: Upper part consisting of a NOT gate and a DFF toggles the XNOR output at every sclk positive edge from the SPI indicating a new LTC bit start. Lower part toggles the XNOR output at every sclk negative edge from the SPI - ONLY IF the data bit stored in the first DFF is '1'.

Audio relay:

Works as a latch being transparent if Audio_Enable is set on next FrameSync positive clock edge. Primarily used for suppressing corrupt data from being transmitted during power up.

LTC lock checker:

The Counter_1 will count 80 bytes sent from the SPI and then set the comp output.

If DelayedFrame is high at this time, isr_Locked will be trigged. Also the bit Status_Reg_1[status_1] will indicate that the LTC data is locked to the delayed frame sync signal.

Status reg:

Bit0: High if SPI is out of data.
Bit1: High if LTC stream is locked to DelayedFrame.

Phase timing:

PWM_3 is a fast one shot generator trigged by DelayedFrame. This one shot will reset the µs counter Counter_2, which is then captured by the first bit in the LTC data stream causing isr_PhaseTiming to be trigged. The value of the Counter_2 will hence contain the µs difference in phase between DelayedFrame and LTC bit #1. The firmware will use this value to adjust PWM_1 period in the digital PLL.

Schematic sheet: LTC analog

The bi-phase modulated digital LTC signal is guided into an integrator to provide proper rise/fall time of 40 µs according to LTC specifications. Note that the integrator is only controlling the slew rate during virtual zero condition. For every pulse it will be forced into saturation and it's virtual zero will be operated rail-to-rail. This means we need to take approx. 15 µs from the delayed frame sync pulse to discharge the integrator's capacitor before entering its linear region (see Frame sync schematic sheet).

Schematic sheet: RTC & VITC

An I²C master communicated with the external RTC and VITC circuits.

Schematic sheet: GPS

A pulse per second (PPS) signal is read from the GPS board.

Schematic sheet: Timers

Slow timers used internally.

Schematic sheet: I²C & UART

The I²C slave component takes care of the communication with the mainboard and the UART is for debugging only.

Schematic sheet: IOs

Collection of pins on the µP which should be connected to ground for limiting noise impact on the board. Also contains the main digital input clock source from the mainframe's OCXO 27 MHz.

Schematic sheet: Analog PLL (not used, but active)

The output from the external PLL circuit is here divided by 160 end fed back into the external PLL circuit's 'compare in', resulting in an output frequency around 4 kHz (PAL 25 fps * 160 = 4000). The PLL lock detector checks if the external PLL is locked to the delayed FF_G or FF_M signal.

Schematic sheet: FF simulator (disabled)

Disabled sheet, used for debugging the board as stand alone. The FF simulator can provide artificial frame sync pulses.

Schematic sheet: UBLOX (disabled)

Disabled sheet, used for future direct UART connection to the GPS chip on PT8616.

Time sources

The PT8620 starts up in PAL, RTC mode. The mainboard handles all changes between video systems and time sources.

RTC

The µP will read the RTC counter and convert it into the current time. The first LTC data is sent out on the next frame sync pulse as HH:MM:SS:01 for PAL. For NTSC the first frame number can be 01 or 03 due to drop frame procedures. The µP now acts as an RTC clocked by the OCXO and phase locked to the frame sync pulses to be genlocked to the rest of the system. The RTC can be set from the mainboard user interface (display, keyboard, serial). If GPS is used as time source, a valid GPS time will update the RTC counter automatically.

GPS

The mainboard tells the PT8620 to use GPS as time source by writing binary x0010xxx to register 3 (see Table 1). The PT8620 µP will wait for a pulse per second (PPS) rising edge from the GPS board. Then it will ask the mainboard for the current GPS time, by setting bit 7 (GPS req) in register 24. The mainboard responds by copying the GPS time string into registers 4-14 and setting bit 3 (set time) in register 3. This response is expected within 400 ms so the fetched GPS time will correspond to the PPS signal earlier received. It is assumed, that the GPS will report a time related to a PPS signal 500ms before and after the PPS . If the response is received later than 400 ms after the PPS, the bit 7 (GPS req) of register 24 will be cleared and set again to start a new GPS time search sequence. If the GPS time search sequence is successful, the PT8620 will add one second and one frame to the time and wait for the next PPS before the LTC data stream is started. E.g. If the time "22:00:00" is received, then the first LTC data to be generated is "22:00:01:01". A valid GPS time will update the RTC automatically. As from firmware version 1.5 any request for a GPS time will start a resync command followed by a 45 seconds pause allowing PT8616 to update its internal RTC. This is required because the requested time will come from PT8616 RTC and not directly from the GPS chip.

VITC

The µP will read the VITC data from one frame (two fields), add one frame and send it out as LTC data on the next frame sync pulse. When generating LTC data from VITC, the sync generator should be genlocked to the incoming video signal.

Note on functionality

The LTC telegram is clocked by the mainframe's 27MHz clock and frequency tuned to each frame according to LTC specifications. If the frame sync pulse jumps for any reason e.g. NTSC resync, the LTC telegram will not be fixed to the frame but will tune in on the nearest frame sync pulse a.s.a.p. typically within one minute. This is done by adjusting the LTC clock rate slightly, allowing an LTC decoder to continue reading telegrams within LTC specifications.

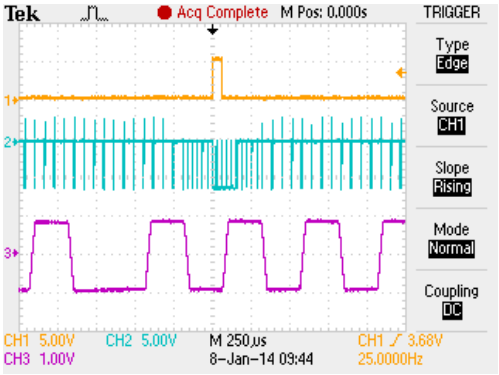
I²C mainboard interface

I²C registers on the LTC-module, address 0x55						Version 1.5					
Register		**direction**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**bit**
0	set hh	rw	h	h	h	h	h	h	h	h	
1	set mm	rw	m	m	m	m	m	m	m	m	
2	set ss	rw	s	s	s	s	s	s	s	s	
3	ctrl1	rw	-PAL/NTSC	Time SRC2	Time SRC1	Time SRC0	Set Time	Refresh	Non-drop		
4	*	rw									
5	[ ]	rw									
6	HH	rw									
7	HH	rw									
8	:	rw									
9	MM	rw				Time string from GPS					
10	MM	rw									
11	:	rw									
12	SS	rw									
13	SS	rw									
14	/0	rw									
24	ctrl2	r	GPS req	PPS OK	GPS gen	PPS	VITC OK				
25	get hh	r	h	h	h	h	h	h	h	h	
26	get mm	r	m	m	m	m	m	m	m	m	
27	get ss	r	s	s	s	s	s	s	s	s	
28	ID	r	1	0	1	0	1	0	1	0	
29	fw version	r	msb	msb	msb	msb	msb	msb	msb	msb	
30	fw version	r	lsb	lsb	lsb	lsb	lsb	lsb	lsb	lsb	
ctrl1	(mainboard to LTC module)										
bit 7	-PAL/NTSC		Video system: 0=PAL, 1=NTSC								
bit 6 *bit 5* *bit 4*	Time SRC2 Time SRC1 Time SRC0		Time source: 000=RTC, 001=GPS, 010=VITC								
bit 3	Set Time		1=Set LTC time (including the onboard RTC) by registers 0-2. Cleared by LTC-module								
bit 2	Refresh		1=Refresh. Refresh time from the current source. Used for daylight savings and NTSC resync. Cleared by LTC-module								
bit 1	Non-drop		0=Drop frames, 1=Non drop frames								
ctrl2	(LTC module to mainboard)										
bit 7	GPS req		LTC sets this to 1 if it wants mainboard to provide GPS time in registers 0-2								
bit 6	PPS OK		LTC sets this to 1 if the GPS pulse per second (pps) is received periodically								
bit 5	GPS gen		LTC sets this if successfully genlocked to GPS								
bit 4	reserved		Reserved								
bit 3	VITC OK		LTC sets this to 1 if VITC is received on video input								

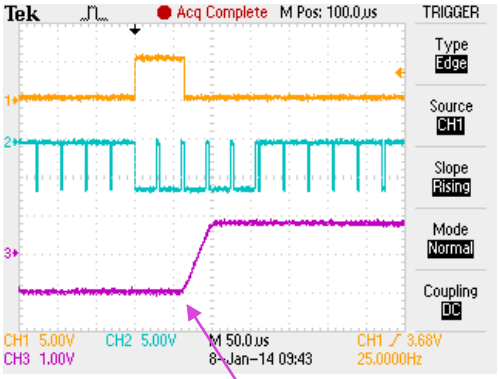
Table 1: I²C mainboard interface

Signal verification

PAL

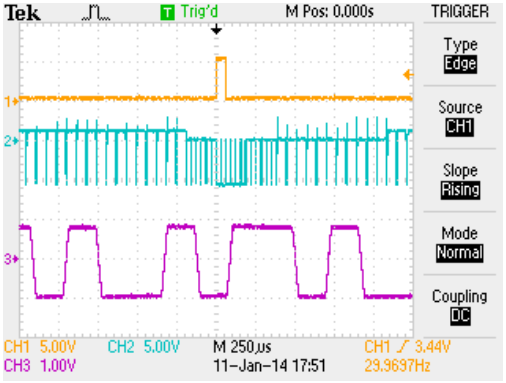


CH1: FF_G internal PAL frame sync pulse
CH2: PAL Black Burst output
CH3: LTC data out (one channel)

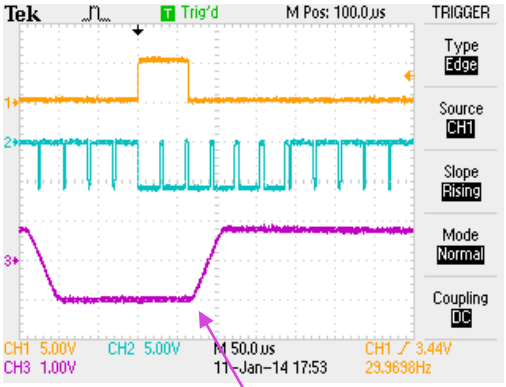


Zoomed in. FF_G pulse is in sync with line #1 on the BB output. First bit of LTC stream starts 64 µs later with a rise time of 40 µs.

NTSC



CH1: FF_M internal NTSC frame sync pulse
CH2: NTSC Black Burst output
CH3: LTC data out (one channel)



Zoomed in. FF_M pulse is in sync with line #4 on the BB output. First bit of LTC stream starts 64 µs later with a rise time of 40 µs.

The end of the LTC signal is easily recognized by a large number of 1's followed by a single 01. The last 1 is almost done when the FF_x signal triggers the scope.

Testpoints

TP1: PLL Locked. This test point is high if the 4 kHz signal is locked to the delayed frame sync pulses.

TP2: Unbalanced LTC output.

TP3: 4 kHz signal locked to delayed frame sync.

TP4: +12V

TP5: -5V

TP6: +5V

TP7: +1.8V

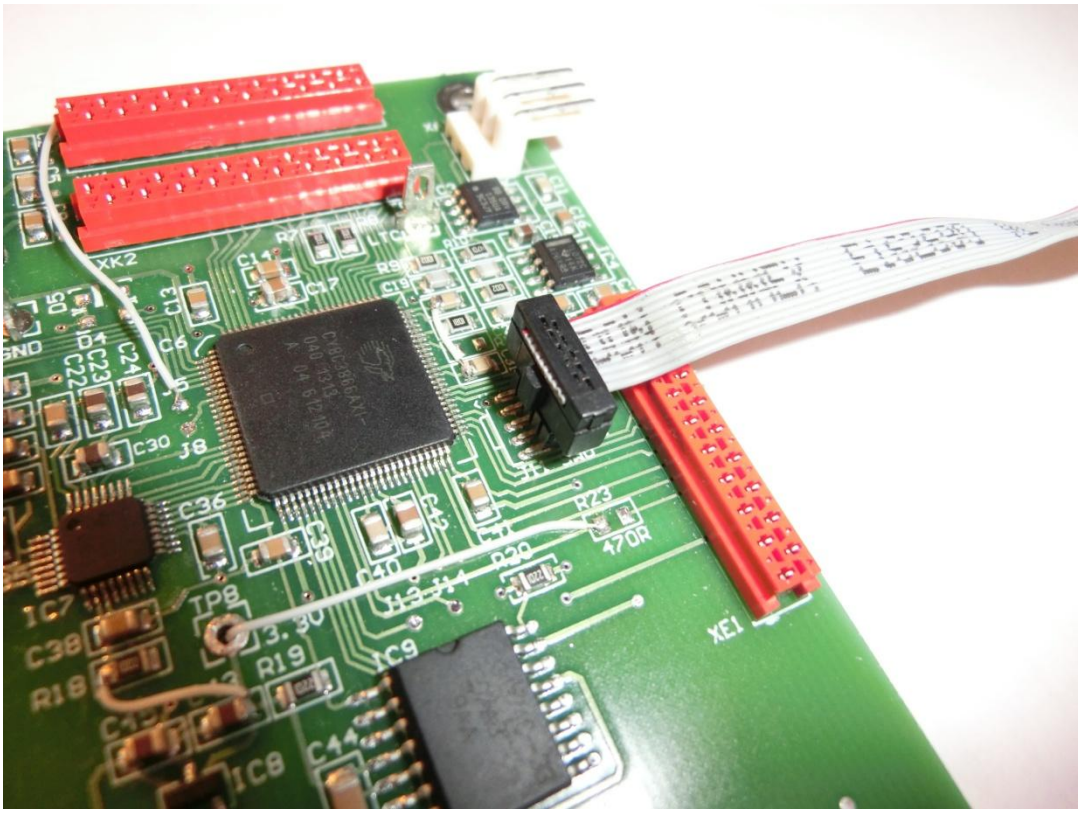
TP8: +3.3V

Programming

Programming procedure:

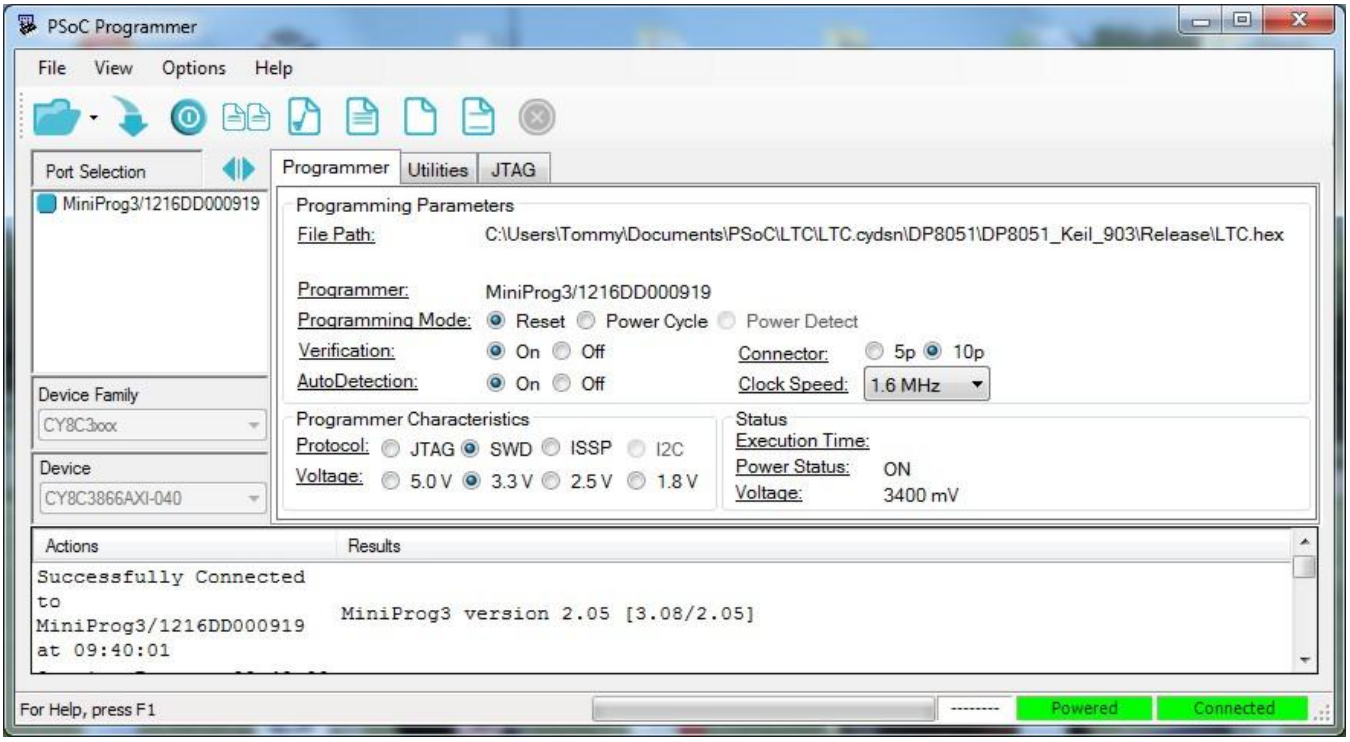
1) Connect the LTC board to the mainboard via the XK1 terminal.

2) Connect the Cypress MiniProg3 programmer to the J11 terminal as shown below.



3) Run the program PSoC Programmer (supplied with MiniProg3 or www.cypress.com).

4) Set up the PSoC Programmer as shown in the next figure. Do NOT apply power from the programmer to the LTC module which should be powered by the VariTime Sync Generator mainboard only.



5) Power up the VariTime Sync Generator.

6) Open the provided hex file, e.g. "LTC.hex".

7) Program the board.

8) Shut power off for the VariTime Sync Generator.

9) Unplug the programming cable and restore VariTime Sync Generator power to initiate a fresh restart.