

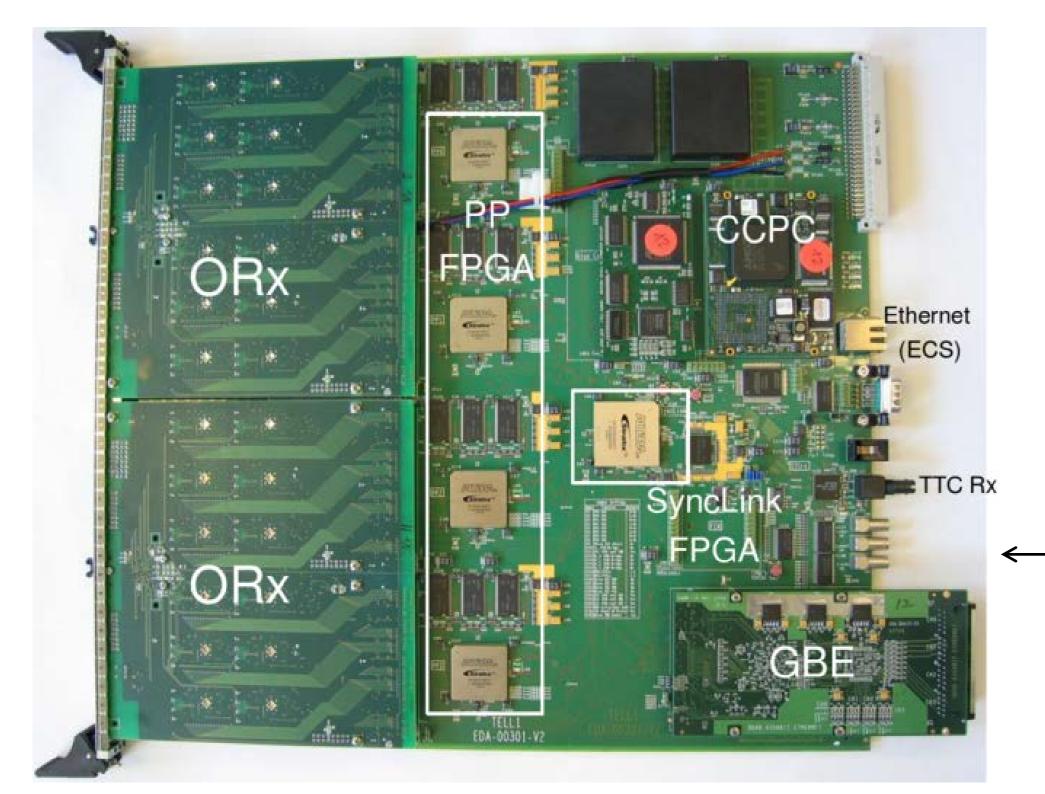
A framework for hardware integration in the LHCb Experiment Control System

LHCD

Close

L. Granado Cardoso, C. Gaspar, R. Schwemmer, J. Barbosa, F. Alessio, CERN, Geneva, Switzerland P-Y. Duval, CCPM, Marseille, France

LHCb is an LHC experiment and collects data from the collisions. In order to acquire the data, LHCb has an infrastructure composed of several sub-detectors to record different parameters of the events.



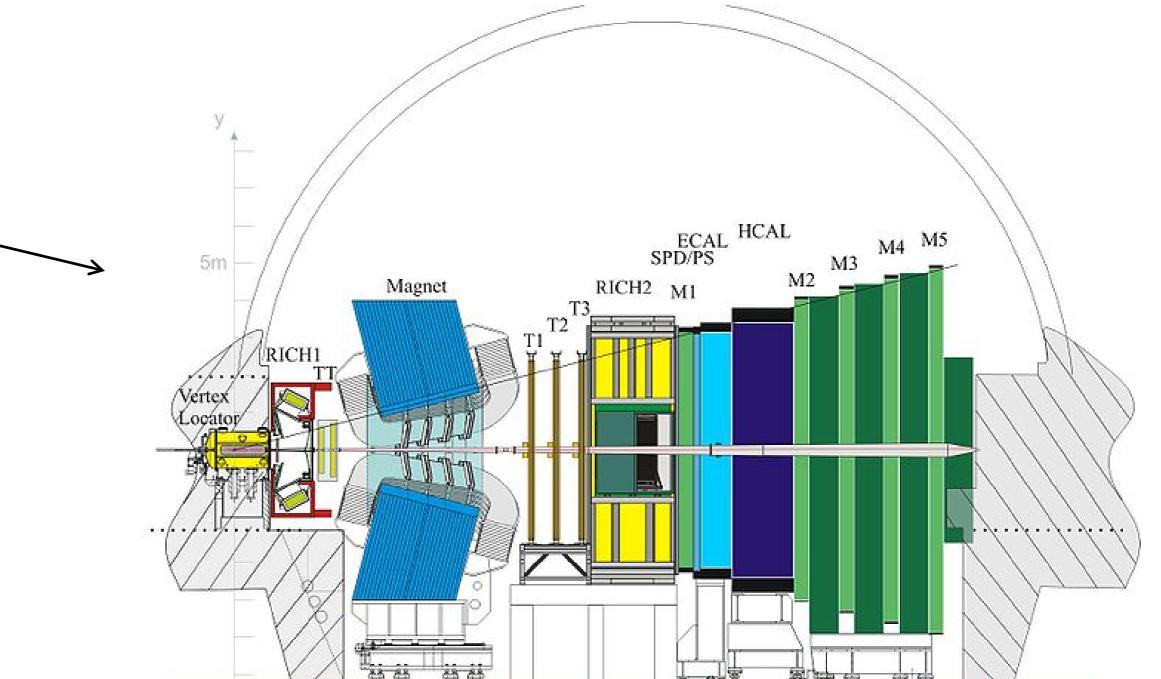


Fig 2. Custom electronic board for data acquisition (TELL1)

10m 5m 15m

Fig 1. LHCb and its sub-detectors

Each sub-detector has specialized custom electronic boards and devices to acquire the data from the events produced in the detector. These devices need to be integrated in the LHCb Experiment Control System (ECS) in order to be easily controlled and monitored.

These devices have a big number of registers and also interface other devices which can be accessed by several different protocols (I2C, Parallel Bus, etc.) each with different types of settings.

	D:\tell1_st.xml - Notepad++	- 🗆 ×	🔅 HW Tool: HW Tool (TDDAQ -	TDDAQ; #1) (on dumdaq01) _ 🗆			
<u>File Edit Search View Encoding Langu</u>	age Se <u>t</u> tings <u>M</u> acro <u>R</u> un <u>P</u> lugins <u>W</u> indow <u>?</u>	X					
	ə 🖒 🛍 🍇 🔍 🔍 📴 🔤 🗉 1 🎼 🐼 💹 🖉 🔍 🖉 🔍 🗷		TYPE HARDWARE ADVANCED				
			ITTEL HARDWARE ADVANCED	1	🌼 🛛 🔅 Registers View (TDDAQ	2 - TDDAQ; #1) (a	n dumdaq01)
tell1_st xml							
1 xml version="1.0" enco</td <td></td> <td></td> <td>Hardware types</td> <td>Registers (hardware type: MONITOR)</td> <td>Liensburger, durcht IIO1 (Liest aufer ein 10.05.17)</td> <td></td> <td></td>			Hardware types	Registers (hardware type: MONITOR)	Liensburger, durcht IIO1 (Liest aufer ein 10.05.17)		
<pre>2 <!--XML HW Description f 3</td--><td>or type "IEDDI_SI'></td><td></td><td></td><td></td><td>Hardware dumtell01 (Last refresh: 10:35:17)</td><td></td><td></td></pre>	or type "IEDDI_SI'>				Hardware dumtell01 (Last refresh: 10:35:17)		
4 <pre>4 </pre> <pre>4 </pre> <pre>4 </pre> <pre>4 </pre> <pre>CCPC" name</pre>	e="GBE">		+ TELL1_OT (type: CCPC) + TELL1_PSSPD (type: CCPC)	ConstantReg (type: LBUS)	REGISTER	TYPE	DATA
5 <node n<="" td="" type="GBEPORT"><td></td><td></td><td>+ TELL1_RICH (type: CCPC)</td><td>EvtAssemCntReg (type: LBUS) PPTriggerCntReg (type: LBUS)</td><td></td><td></td><td></td></node>			+ TELL1_RICH (type: CCPC)	EvtAssemCntReg (type: LBUS) PPTriggerCntReg (type: LBUS)			
6 <node n<="" td="" type="GBEPORT"><td></td><td></td><td>- TELL1_ST (type: CCPC)</td><td>PPBankCntReg0 (type: LBUS)</td><td>TDDAQ:dumtell01.files.jam</td><td>String</td><td></td></node>			- TELL1_ST (type: CCPC)	PPBankCntReg0 (type: LBUS)	TDDAQ:dumtell01.files.jam	String	
│	ame="PORT2">		PROM (type: EEPROM) - TEMP (type: TEMP)	PPBankCntReg1 (type: LBUS) PPEventCntReg (type: LBUS)	TDDAQ:dumtell01.files.pof	String /hom	e/niko/tell1/tell1_ST_v4.1.1.pof
<reg address<="" size="1" td=""><td>="0x0000018" type="GBE" name="<u>Config</u>" onchange="0" poll="2"/></td><td></td><td>T1FPGA (type: TEMPSENS)</td><td>InfoParityErrorCntReg (type: LBUS)</td><td>TDDAQ:dumtell01.files.cfg</td><td>String /hom</td><td>e/rainer/ST3.v23.cfg</td></reg>	="0x0000018" type="GBE" name=" <u>Config</u> " onchange="0" poll="2"/>		T1FPGA (type: TEMPSENS)	InfoParityErrorCntReg (type: LBUS)	TDDAQ:dumtell01.files.cfg	String /hom	e/rainer/ST3.v23.cfg
<reg address<="" size="1" td=""><td>="Ox0000002" type="GBE" name="Duplex" onchange="0" poll="2"/></td><td></td><td>T2GBE (type: TEMPSENS) TTCRX (type: TTCRX)</td><td>PPErrorMonReg (type: LBUS)</td><td></td><td></td><td></td></reg>	="Ox0000002" type="GBE" name="Duplex" onchange="0" poll="2"/>		T2GBE (type: TEMPSENS) TTCRX (type: TTCRX)	PPErrorMonReg (type: LBUS)			
	="0x0000010" type="GBE" name="MAC" onchange="0" poll="2"/>		- PPO (type: PPST)	PPEcsErrorCntReg (type: LBUS) PPDerandEvntCntReg (type: LBUS)	TDDAQ:dumtell01.GBE.PORT3.RXStat_Transform	mei DynInt 0 0	50510874 0 0 0 0 0 0 0 0
	="0x0000060" type="GBE" name="PHYControl" onchange="0" poll="2"/>		CMNCTRL (type: CONTROL)	PPDateReg (type: LBUS)	TDDAQ:dumtell01.GBE.PORT0.RXStat_Transform	me DynInt 0101	725701196 0 0 0 0 0 0 0
	="0x0000061" type="GBE" name="PHYStatus" onchange="0" poll="2"/>		CMNMON (type: MONITOR) CMNMEM (type: RAM)	PPTimeReg (type: LBUS) PPVersionReg (type: LBUS)			
	<pre>g="0x0000020" type="GBE" name="RXStat" onchange="0" poll="2"/> pre="DXGtat "pre=Gamed"/></pre>		ORX (type: ORX)	FIFO_0 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT2.RXStat_Transform	me Dynint 0101	725822258 0 0 0 0 0 0 0 0
	me="RXStat_Transformed"/>		SPCREG (type: SPCFC_REG_ST)	FIF0_1 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT1.RXStat_Transform	me DynInt 0 0	726654322 0 0 0 0 0 0 0 0
	="0x0000600" type="GBE" name="TXFifoHighWtrmrk" onchange="0" poll="2"/> ="0x000060A" type="GBE" name="TXFifoLowWtrmrk" onchange="0" poll="2"/>		SPCMEM (type: SPCFC_RAM_ST) - PP1 (type: PPST)	FIF0_2 (type: LBUS) FIF0_3 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT3.TXStat_Transform	ner Dynint 2288	0 -553493722 141 0 0 0 0
	="0x00000614" type="GBE" name="TXFifoThreshold" onchange="0" poll="2"/>		CMNCTRL (type: CONTROL)	FIF0_4 (type: LBUS)			
	s="0x0000040" type="GBE" name="TXStat" onchange="0" poll="2"/>		CMNMON (type: MONITOR) CMNMEM (type: RAM)	FIFO_5 (type: LBUS) FIFO_6 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT1.TXStat_Transform	mec DynInt 2288	0 -560123304 141 0 0 0 0
	me="TXStat_Transformed"/>		ORX (type: ORX)	FIFO_7 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT2.TXStat_Transform	nec DynInt 3374	0 -284774476 327 0 0 0 0
-	_		SPCREG (type: SPCFC_REG_ST) SPCMEM (type: SPCFC_RAM_ST)	FIFO_8 (type: LBUS)			
<pre> <node n<="" pre="" type="GBEPORT"></node></pre>	ame="PORT3">		- PP2 (type: PPST)	FIF0_9 (type: LBUS) FIF0_10 (type: LBUS)	TDDAQ:dumtell01.GBE.PORT0.TXStat_Transform	nec Dynint 33/4	0 -80096128 327 0 0 0 0
<reg address<="" size="1" td=""><td>="0x0000018" type="GBE" name="<u>Config</u>" onchange="0" poll="2"/></td><td></td><td>CMNCTRL (type: CONTROL)</td><td>FIFO_11 (type: LBUS)</td><td>TDDAQ:dumtell01.GBE.Clock</td><td>GBE ???</td><td></td></reg>	="0x0000018" type="GBE" name=" <u>Config</u> " onchange="0" poll="2"/>		CMNCTRL (type: CONTROL)	FIFO_11 (type: LBUS)	TDDAQ:dumtell01.GBE.Clock	GBE ???	
<reg address<="" size="1" td=""><td>="Ox0000002" type="GBE" name="Duplex" onchange="0" poll="2"/></td><td></td><td></td><td></td><td>TDDAQ:dumtell01.GBE.JTAGID</td><td>GBE ???</td><td></td></reg>	="Ox0000002" type="GBE" name="Duplex" onchange="0" poll="2"/>				TDDAQ:dumtell01.GBE.JTAGID	GBE ???	
	="Ox0000010" type="GBE" name="MAC" onchange="0" poll="2"/>		CMNMON		· ·		
	="0x0000060" type="GBE" name="PHYControl" onchange="0" poll="2"/>				TDDAQ:dumtell01.GBE.MACSoftReset	GBE ???	
	="Ox0000061" type="GBE" name="PHYStatus" onchange="0" poll="2"/>		Create Remove Insert	Create Remove Massive	TDDAQ:dumtelI01.GBE.MDIOControl	GBE ???	
	<pre>g="0x0000020" type="GBE" name="RXStat" onchange="0" poll="2"/> pre="DXGtat "pre=Gamed"/></pre>					GBE ???	
	me="RXStat_Transformed"/> ="0x00006600" type="GBE" name="TXFifoHighWtrmrk" onchange="0" poll="2"/>		Subscribe Unsubscribe Refresh	Show registers of type (All)	TDDAQ:dumtell01.GBE.Mode	GBE ///	
	="0x000060A" type="GBE" name="TXFifoLowWtrmrk" onchange="0" poll="2"/>				TDDAQ:dumtell01.GBE.PHYCtrl	GBE ???	
	="0x0000614" type="GBE" name="TXFifoThreshold" onchange="0" poll="2"/>		Generate				
	s="0x0000040" type="GBE" name="TXStat" onchange="0" poll="2"/>		Evenent Bessine Turses				
	me="TXStat_Transformed"/>		Export Recipe Types				
	-						
<node na<="" td="" type="EEPROM"><td>me="PROM"/></td><td></td><td></td><td></td><td>Manifarina al Data I</td><td></td><td></td></node>	me="PROM"/>				Manifarina al Data I		
6 <reg address="</td" size="1"><td>"0x0000794" type="GBE" name="Clock" onchange="0" poll="2"/></td><td></td><td></td><td>v4 r5</td><td>Monitoring Recipes</td><td>Refr</td><td>resh Clo</td></reg>	"0x0000794" type="GBE" name="Clock" onchange="0" poll="2"/>			v4 r5	Monitoring Recipes	Refr	resh Clo
	"0x000050C" type="GBE" name="JTAGID" onchange="0" poll="2"/>						
	"0x0000505" type="GBE" name="MACSoftReset" onchange="0" poll="2"/>				1		
_	"0x0000683" type="GBE" name="MDIOControl" onchange="0" poll="2"/>		HW Para	Close			
	"0x0000501" type="GBE" name="Mode" onchange="0" poll="2"/>						
<pre>//l //////////////////////////////////</pre>	IDV0000680U type=UCURU news=UDUVCtrlU enchange=UOU nell=UOU/N						

fwHw is a tool that allows the abstraction of these electronic devices and create models of the

hardware. This hides the complexity of accessing the hardware and allows for an easy integration into the ECS.



Fig 3. XML File with the hardware description (TELL!)

Fig 4. HW Tool with the imported model from the XML File and some of the defined named registers

From an XML file containing the description of the hardware and all the settings necessary to access the hardware registers, a model can be created in the control system. Registers are then able to be accessed via their defined names. These settings are transmitted to a server that interfaces the hardware and is aware of all the different protocols to communicate with the hardware.

> The abstraction of the hardware into models, also allows for the usage of recipes (named sets of configurations) which can be applied to the registers of the devices. This allows to easily configure the devices for various running modes of the LHC.

Subdetectors can now create their own User Interfaces and control trees. They will use libraries provided by the fwHw tool to access their hardware via the defined registers in the models.

pe		Hardware myTell1				
		Values last written to hardw	are:			
ardware myTell1	REGISTER	TYPE	DATA	1		
Values last written to hardware:		myTell1.PROM.Id	12C	FEEDBABE		
REGISTER TYPE		myTeli1.TTCRX.FineDelay1	12C	ff		
myTell1.PROM.Id	12C	myTell1.TTCRX.FineDelay2	12C			
myTell1.TEMP.T1FPGA.TempRead	12C	myTell1.TTCRX.CoarseDelay	120			
myTell1.TEMP.T2GBE.TempRead	12C			0		
myTell1 ARX DACPP0.BurstWrite	12C					
myTell1.ARX.DACPP0.Reset	12C					
myTell1.ARX.DACPP1.BurstWrite	12C					
myTell1.ARX.DACPP1.Reset	12C		_			
myTell1.ARX.DACPP2.BurstWrite	12C			-		
myTell1.ARX.DACPP2.Reset	12C					
myTell1.ARX.DACPP3.BurstWrite	12C					
myTell1.ARX.DACPP3.Reset	12C					
myTell1.TTCRX.IACId	12C		_			
myTell1.TTCRX.I2CId	12C				~	
myTell1.TTCRX.FineDelay1	12C					
myTell1.TTCRX.FineDelay2	12C	Recipe Name:				
myTell1.TTCRX.CoarseDelay	12C	recipieNr1	•	Save		
myTell1.TTCRX.Control	12C	Choose Recipe Type:				
		TELL1_VELO_TEST	•	Ok Cancel	Clos	
				and the second second	63	
ecipe Name:						
recipieNr1	•	Save				
hoose Recipe Type:						
1 21	-	Ok Cancel	Close			

Fig 5. HW tool recipes for some registers of a device

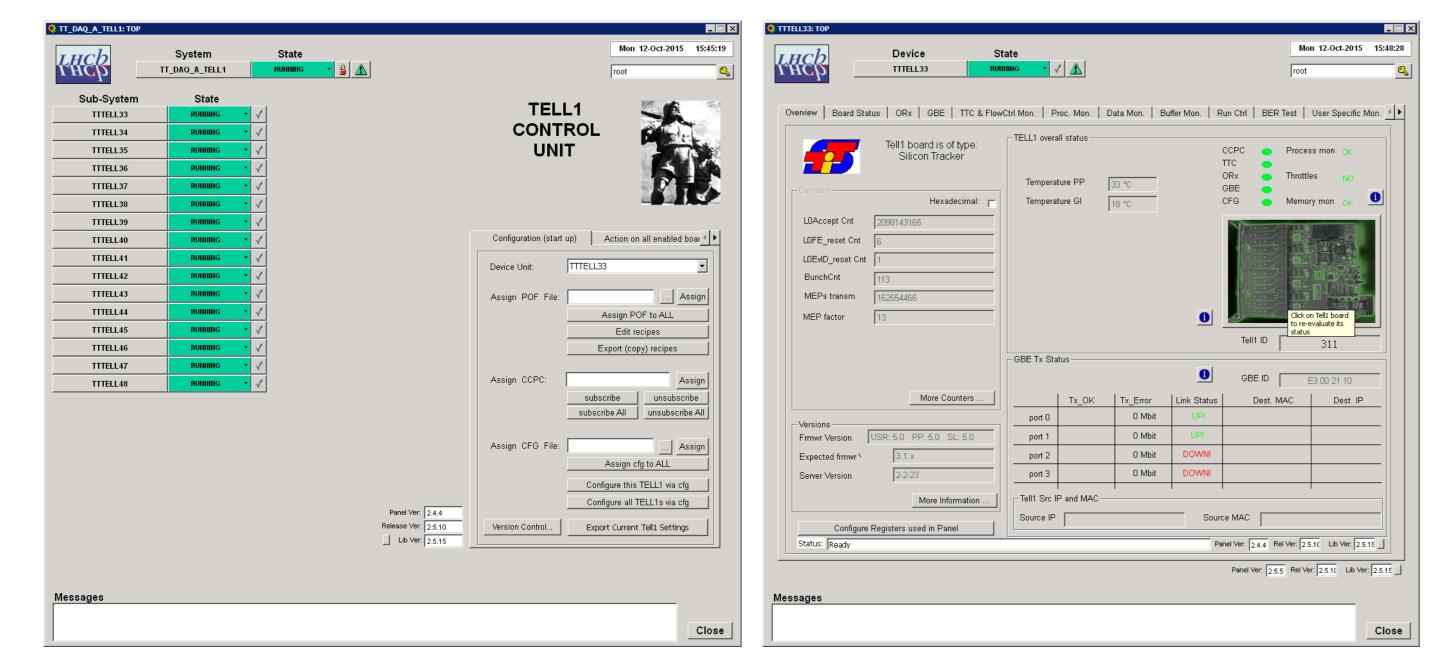


Fig 6. Control Tree and TELL1 panel for one of the LHC sub-detectors. Communication with the hardware is implemented via fwHw libraries

The fwHw tool removes complexity from the system, by abstracting the models of the devices into the Control System. This hides the complexity of the hardware access, allowing the communication with the hardware registers to be done just using the names defined in the models. It provides libraries that sub-detectors can use to easily create the specific panels and control trees for their specific devices. It also allows for the usage of recipes for easier configuration of the devices and as a base for automatic actions on the ECS.

ICALEPCS, Melbourne, October 2015