

## **Philips TinyARM DIP40 Module**

### **1. Introduction**

The purpose of this document is to specify mechanical dimensions and electrical behaviour of TinyARM LPC2106 DIP40 module.

### **2. Reason for development**

Philips introduced ARM processor housed in LQFP 48 pin package. It features 128kbyte Flash memory and up to 64kbyte SRAM. This chip can be used as a replacement for 8/16 microcontrollers.

There are following problems when using this chip:

- A. Price of development tools, both software and hardware.
- B. Unusual package which is not convenient for development.
- C. The chip requires 2 different voltages.

### **3. Solution**

TinyARM DIP40 module intends to solve all problems:

- A. TinyARM module is housed in DIP40 package which has pinout similar to 8751 microcontroller.
- B. TinyARM requires just one convenient 5V supply. All the voltages required for Philips processor are generated by TinyArm module.
- C. TinyARM module features built-in 10 MHz oscillator.
- D. TinyARM module includes Reset generator.
- E. TinyARM is re-usable. It can be used in several designs, the only additional investment is DIL40 socket for each design.
- F. Software tools are either IAR or free GNU .

### **4. Development version of TinyARM module**

To be defined later.

### **5. List of required signals for development version**

All the signals required for standard ARM Test/Debug JTAG interface are generated on JtagTinyARM board. This includes standard JTAG TRSTn, TCLK, TMS, TDI and TDO signals and handling of RTCK and DBGSEL signals. Reset signal for all circuitry is generated on board.

### **6. Voltage levels and electrical behaviour**

TinyARM Module runs from 5V supply applied to DIL40 pin 40. The supply voltage for this version can be from 4.5V to 5.5V. The estimated current consumption is 100mA from 5V supply.

TinyARM module can supply 3.3V voltage from its "virtual" pin 42. Pin 42 is positioned on the right side of DIP40 pin 1 and is marked as "3V". Do not draw more than 20mA from this supply !

Voltage levels for ARM I/O pins are 3.3V compliant. The specs says that I/O pins are 5V tolerant.

### **7. Method of programming internal Flash**

There are 2 methods to program Flash - either JTAG or ISP (using UART0 of Philips LPC210x chip). Please, refer to the relevant Philips LPC210x manual.

Note: P0.14 must be logic low for TinyARM to enter bootloader mode.

For normal operation, use pullup resistor to Vcc so that TinyARM does not enter bootloader mode.

### **8. Mechanical dimensions**

The module should fit into the standard 600 mils DIL40 socket.

### **9. Tips and tricks for development**

P0.14 must be logic low for TinyARM to enter bootloader mode.

For normal operation, use pullup resistor to Vcc so that TinyARM does not enter bootloader mode.

See also additional notes - fig.16.

### **10. Preliminary drawing**

None.

### **11. TinyARM's board control signals**

**RSTN** - input for external reset signal, active low. Leave unconnected if not used. TinyARM board generates proper internal Reset signal upon power-up. An external push button can be connected to RSTN to generate reset signal. No debounce is needed for the push button.

Note: If driving RSTN pin by external circuitry, then it must be open collector (open drain) with pull-up resistor. Do NOT use push-pull to drive RSTN !

**RST\_OUTN** - output of TinyARM internal reset generator, active low, open drain output.

**PWRENN** - input controlling TinyARM voltage regulators. Default is low when the voltage regulators are enabled and generate 3.3V and 1.8V respectively. This signal is pulled down on TinyARM board. Normally leave this pin unconnected (N.C.). Driving this pin high will disable voltage regulators.

**DBGSEL** - this is Debug Select input. Default is low when the part operates normally, executing code in internal Flash or SRAM. Leave this pin unconnected for normal operation as it is pulled down by 10k resistor on TinyARM module. Drive this pin high if there is a need to enter Debug Mode.

**RTCK** - leave this pin unconnected for normal operation.

### **12. Interfacing TinyARM board to various voltage levels**

TinyARM board runs from 5V supply and generates 3.3V voltage for I/O pins logic levels reference. So all TinyARM I/Os are 3.3V compliant, meaning they can directly interface with 3.3V logic. They can also interface directly 5V LV TTL and HCT MOS logic.

TinyARM I/Os are 5V resistant so they can receive 5V TTL/HC/HCT logic levels. However, I/O cannot drive 5V HC MOS inputs directly, some level conversion is necessary. The simplest method for slow signals is using pullup resistor to 5V supply.

### **13. Simple test / First use of TinyARM**

It is easy to start with TinyARM. All what you need is an experimental board, DIL40 socket, RS232 interface circuit like MAX232 and 5V power supply. Philips supplies LPC210x ISP utility for Windows which can talk to TinyArm's UART0 serial interface.

Make the connections like the following:

- connect ground to DIL40 pin 20
- connect +5V to DIL40 pin 40
- connect logic low (pull down 3k9 to ground) to DIL40 pin 16 which corresponds to TinyARM Port14. This pin is samples during reset or power-up to indicate whether to start ISP utility.
- connect DIL40 pin 1 which corresponds to TinyARM Port00/TxD0 to the input of MAX232 driver, e.g. to Tlin (MAX232 pin 11). The output of MAX232 driver, e.g. Tlout (MAX232 pin 14) is connected to DB9 or DB25 connector as TxD signal.
- connect DIL40 pin 2 which corresponds to TinyARM Port01/RxD0 to the output of MAX232 receiver, e.g. to Rlout (MAX232 pin 12). The input of MAX232 driver, e.g. Rlin (MAX232 pin 13) is connected to DB9 or DB25 connector as RxD signal.
- Connect a push-button between DIL40 pin 9 RSTN and ground.
- Insert TinyARM board into DIL40 socket.
- Connect RS232 cable to the PC on one side and to your board on the other side.
- Apply power from 5V supply.
- Start Philips LPC210x utility
- Press push button to connect TinyARM to the PC, if necessary.
- ENJOY !

### **14. Connecting Macraigor Systems WIGGLER to TinyARM board**

The easiest way is to use extra DIL40 socket and connect DIL40 pins to the male dual-row 20 pin header as follows:

DIL40 pin	Header pin	Signal name
20	4,6,8,10,12,14 16,18,20	GND
22	3	TRSTn
23	7	TMS
24	9	TCK
25	5	TDI
26	13	TDO
9	15	RESETn
30	11	RTCK
42	1,2	TVcc (3.3V)

Some additional resistors are strongly recommended to apply correct default voltage levels on JTAG pins.

Notes:

- a. DBGSEL pin should be driven high, i.e. pullup resistor 1k to Vcc for the part to enter debug mode when debugging is possible.
- b. TinyARM module can supply 3.3V voltage from its "virtual" pin 42. Pin 42 is positioned on the right side of DIP40 pin 1 and is marked as "3V".

**15. TinyARM board pinout**

Pin Nr.	Pin Name	Altern.		Pin Nr.	Pin Name	Altern.
1	P0.0	TxD0/PWM1		40	VCC	5V
2	P0.1	RxD0/PWM3		39	P0.31	EXTIN0/*
3	P0.2	SCL/CAP0.0		38	P0.30	TRACEPKT3/*
4	P0.3	SDA/MAT0.0		37	P0.29	TRACEPKT2/*
5	P0.4	SCK/CAP0.1		36	P0.28	TRACEPKT1/*
6	P0.5	MISO/MAT0.1		35	P0.27	TRACEPKT0/*
7	P0.6	MOSI/CAP0.2		34	P0.26	TRACESYNC
8	P0.7	SSEL/PWM		33	P0.25	PIPESTAT2
9	RSTN			32	P0.24	PIPESTAT1
10	P0.8	TxD1/PWM4		31	RSTN_OUT	
11	P0.9	RxD1/PWM6		30	RTCK	
12	P0.10	RTS1/CAP1.0		29	DBGSEL	
13	P0.11	CTS1/CAP1.1		28	P0.23	PIPESTAT0
14	P0.12	DSR1/MAT1.1		27	P0.22	TRACECLK
15	P0.13	DTR1/MAT1.1		26	P0.21	PWM5/TDO
16 *Note	P0.14	DCD1/EINT1		25	P0.20	MAT1.3/TDI
17	P0.15	RI1/EINT2		24	P0.19	MAT1.2/TCK
18	PWRENN			23	P0.18	CAP1.3/TMS
19	N.C.			22	P0.17	CAP1.2/TRST
20	GND			21	P0.16	EINT0/MAT0.2

Note: P0.14 must be logic low for TinyARM to enter bootloader mode.  
 For normal operation, use pullup resistor to Vcc so that TinyARM does not enter bootloader mode.

**16. Known problems of Philips LPC2106 processor**

- a. External interrupts 0 to 2 (EINT0 to EINT2) are level sensitive. It is not possible to set them to edge sensitive mode.
- b. SPI interface requires to use pullup resistor on SSEL pin even if it is set to Master mode. That results in pin wasting, pin SSEL cannot be used as GPIO P0.7 when SPI is used.