Making First Project Using LadderWORK

OBJECTIVE: MAKING AN LED BLINK CONNECTED AT PORT PIN OF GENERAL 8051 BASED MCU

ASSUMPTION:

✓ DOWNLOADED LADDERWORK EVALUATION SOFTWARE

PREPARATION:

✓ OPEN A NEW PROJET WITH NAME "BLINK"

WRITTEN BY: AJAY BHARGAV

DATED: 21ST NOV, 2006

REVISION CODE: 1.0

Now we are ready to start with the first step to LadderWORK software.

Step1: Adding Components:

In ladderWORK IDE you see a component bar on the right side, which has a number of components to select from.

As we are going to make an LED blink so there must be something which can give u a square wave output. That means "A Clock".

In the component bar select the clock component and u see component attached to your cursor. Now place the component on your worksheet wherever you want.

So you now have something like this on your worksheet.

	lad,					6	9	. 6	8 6		a	0=11	A				
ault								-	1	문. 			F.	ш	De	efaul	t
	0	++	A	-	j 🖲		2	Q	Q.								
				A	6				Ĵ					040	B		
			6 :	*	:::		s::		5 2		:::		5 2		s::		
	- 35	÷	35	Ŷ	35	Ŷ	35	×.	1	÷	35	÷	35	Ŷ	35	2	
	3 2		6 2	*	62	*	62	×	6 :	*	63	*	8 :	*	62	*	
	- 33		3	÷	C	K?	25	×	33	Ŷ	1	÷	1	×	1	Ŷ	
	3 :		8 2	\sim	8 2	Г					8 :	\sim	8 :	*	8 :	*	
	- 35		÷.	×	1	-	лл	лл	Л	÷.	1		1	×	33	÷.	
	- 6 2		6 2	*	6 2		die	ev.		* *	5 2	*	8 :	*	62	*	
	- 33	÷	35	÷	35				10	÷	85	×.	35	×	33	8	
	3 2		6 2	*	62	*	63	×	6 2	\mathbf{x}	5 2	\mathbf{x}	6 2	*	62	*	
	35		35	Ŷ	35	Ŷ	35		35	÷	35		35	×	35	÷.	
	- 62		62	*	62	*	63 :	*	3 2	*	82	*	8 2		62 :		
	35		35	÷.	35	÷.	35	×.	35	÷	35	÷.	35	2	35		
	- 52		62		62		82	*	8 2		82	÷	82	*	62	1	

Step 2: Editing System Configuration:

Before we setup the clock parameters we first need to set the system configuration.

Go to Option menu \rightarrow System Edit

Now you see the System I/O Module Editor Dialogue.

Here we define the configuration for the MCU we are going to use and also configure the port where we are going to blink our LED.

In our case we will select the "MPU8051"

For selecting, Double click on the MPU8051 for general 8051 CPU core with P0 to P3.

So now you have MPU8051 in your current configuration section.

Now selecting that module you can edit the Module configuration. Here you can define the ports to be Input or Output, Logic High or Logic low etc.

Current Configuration		Available I/O M	odules				
Module# Name \$\$#1 MPU8051	Description GENERIC 8	Name SMPU8051 MB00L8 MB00L16	Descp GENERIC 8051 CI BOOL x 8 memory BOOL x 16 memor	PU OR COMPA module v module	TIBLE P	DRTS PO TO	
		Module Configura	ation				
<	•	VO Name	Reference P1_0 P1_1 P1_2 P1_3 P1_4 P1_5 P1_6 P1_7	Type BOOL BOOL BOOL BOOL BOOL BOOL BOOL	Dir OUT IN IN IN IN IN IN	Polarity NEG(-) NEG(-) NEG(-) NEG(-) NEG(-) NEG(-)	Er ^ 0 0
leference Name P1_ Direction Pola Input O (+ O Utput O (-	0 rity Ena +) Normal ()) Negated ()	Phisical Address able Comm nable Disable	20X\$(AL1),1.0	Value		Up	date

As we are going to use Port1 Pin 0 for our LED. Select P1_0 from the module configuration block. Change the direction of the port pin to Output and click update button.

Now you have configured the system for your LED. Click ok and exit from the system I/O module editor.

Step 3: Adding Output Port:

To add an output port select the output from the right component bar and add it in the worksheet.

now added another component in your worksheet. Double-Click on the output component and change its Parameters.

You will see a window like this

- -	-	٨	last 6		00		4 Z	100	srault			
	П	A	1 de 1	19	No line							
			A					В				
1.2												
1	4	÷.	10.00	10.00	1.1	1.1	1.1	\sim \sim	$\mathcal{L}_{i} = \mathcal{L}_{i}$	$\mathcal{L} = \mathcal{L}$	10 14	
2			$\sim \infty$									
1	4	÷.	• 6		1.1	10.0			$\mathcal{L} = \mathcal{L}$		10 - 52	
	23		10.0	10000			10.00	$\sim \infty$	$\sim 10^{-10}$	$\sim 10^{-10}$	$\sim \infty$	
1	14	1	1. 1.	L U	w	. (γ	1.1	1.1		10 14	
2			$\sim \infty$		orx [. (<u></u>		$\sim \infty$	~ 2	100	
1	- 2		10.00			12 2			\sim \sim		10 14	
	- 23	ſ	OUTPU	T (LAD	DER)			10	10.00	16 📻	x	
	2											
3	2		Refere	ence	P1_0				-	Brows	e	
						.						
	- 23		Chann	nel							- 1	
								Can	cel	OK		
3	*	L										
			20.0	10.03	10.03		10.03	10.03	10.03	10.0	200	
3	27			10.0								

From the Drop down reference box, select the port where u want the output be active. As in our case its P1.0 so select P1_0 from the drop down.

Click ok and now connect the other pin of the output component to ground. Selecting ground component from the right component bar.

Set the frequency of clock by double clicking the clock component. Enter the value of frequency you want. Then click ok.

Now our design is complete.

Step 4: Setting Compiler option:

Go to Option menu \rightarrow Compiler

Now to get the hex file ready we need to compile and build our Ladder Design. For that, First we need to configure the compiler options, to tell the compiler how to MAP the code and data segment.

Options compiler		
Code Generation Linker Files. MODB	JS(R) Kernel	Profiles Profile name
8051 Memory Internal 	System Clock Frequency (MHz) 12,000(Default
Temporary Node Memory Always use 8051 Internal Memory Use external when requested Timing Precision Low Medium High Auto Debounce After Phisical Input On On Off	Syncronism Node Memory	Default

Set the configuration according to your board. We don't need much ram for our project so select the following configuration for our project:

- ✓ 8051 Memory: Internal
- ✓ Temporary Node Memory: Always use 8051 Internal Memory
- ✓ System Clock: Set what you are using on your board (12.000MHz in our case)

Now next we need to import profile for our project. In the Profiles block.

Click on the "Import" button. You will see a Select profile Dialogue box.

Select profile "Default 8051 INT RAM" from the Available profiles and click ok button to exit.

Code Generation Linker Files. MODBUS(R) Kernel Solutions compiler Select Clock Profile name Profile name B051 Memory External Internal External Temporary Node Memory Import Use external when requested CPG-50 LARGE Timing Precision ErAULT 8051 EXT RAM Evaluation Board ADuC812 GPC R/T 94 EMULATOR GPC R/T 94 EMULATOR GPC553 EXT RAM GPC553 EXT RAM GPC553 EXT RAM GPC553 EXT RAM GPC553 EXT RAM	A	В	С
Code Generation Linker Files. MODBUS(R) Kernel 8051 Memory Internal External Profile name Profile name Temporary Node Memory Always use 8051 Internal Memory Use external when requested Timing Precision Low Medium High Auto Debounce After Phisical Input Off OK Cancel Profiles Profile name Default Import Update Remove Default BerAult 8051 INT RAM Evaluation Board ADuC812 GPC R/T 34 EMULAT 1005 GPC R/T 34 EMULAT 100H GPC F53 EXT E2PR0M GPC S53 EXT E2PR0M GPC S53 EXT E2PR0M GPC S53 EXT RAM OK Cancel 	ptions compiler		
O O OF OK Cancel	Code Generation Linker Files. MODBL 8051 Memory (a) Internal (b) External Temporary Node Memory (b) Always use 8051 Internal Memory (c) Use external when requested Timing Precision (c) Low (c) Medium (c) High Auto Debounce After Phisical Input	Select Clock Available profiles CPG-50 LARGE CTD 2000 CTD 2000 EXT RAM DEFAULT 8051 INT RAM DEFAULT 8051 INT RAM Evaluation Board ADuC812 GPC R/T 94 FINAL ROM GPC168 EXT RAM GPC553 EXT RAM GPC553 EXT RAM	Profile name Default Import Update Remove Default
	Auto Debounce After Phisical Input	GPC553 EXT RAM	

Now lets configure the linker MAP. Click on linker tab in same Compiler option window and set the linker configuration, How you want to map your output code against the memory.

DEEALILT 8051 INT BAM
Import Update Remove
Default
7F DEFAULT 8051 INT RAM
FFFF
zing

After this click ok. To exit the compiler configuration window. Now we are ready to build our project.

Step 5: Building Project:

To build the project

Go to Build \rightarrow Compile

After the successful compilation of the design u see this prompt.

In the project directory where we have created our project you will find Blink.HEX file.

Load the hex file on to your target board and Run!