Digital Relay Protection with LPC 2124 General Purpose I/O Controller for External Communication

Slavi Yasenov Lubomirov

Abstract – **The paper describes the advantages of digital relay protection with LPC2124. It is proven that their precision and fast performance depends mostly on the microprocessor module. An attempt is made for a simulation of the use of external communication pins for the input signals of the digital relay protection for the LPC-E2124 TCP-IP development board with LPC2124. A detailed description of the possibility for external communication of the hardware module through 31 additional optional GPIO's multiplexed on peripherals is given.**

Keywords – **Protection relay, GPIO, General Purpose I/O ports.**

I. INTRODUCTION

Malfunction of relay protection is one of the major causes for severe breakdowns that happen periodically all over the world in the electrical power supply industry.

The continuous development and improvement of microprocessor devices which we have witnessed recently has inevitably influenced the design of relay protection. The integration of microprocessor controlled automatic devices for malfunction protection includes not only protective turn-off systems, but other automatic functions as well. Such are automatic frequency unbinding, subsequent automatic turn on, automatic backup turn on, etc. All of these are intelligent technical devices for automatic control and, due to the programming flexibility, possess the ability to modify and adapt their settings in accordance with the specific circumstances of the breakdown, as well as the capability for self-testing, self-diagnostics and self-improvement.

Digital relay protection is much more reliable compared to past generations relay protection because of its numerous advantages: much smaller devices which consist of a smaller number of components with significantly lower rate of wearing out, a small number of internal joints, high speed information processing, constant monitoring of input measurement data, analog-to-digital data conversion, measurement data transmission for further processing and analysis, etc.

It must be noted that digital relay protection devices can include several microprocessors each of which is connected to or responsible for accomplishing a separate portion of the task at hand thus ensuring efficient high-speed protection.

There are a lot of common characteristics in the existing relay protection modules and their block diagrams are similar.

Slavi Lyubomirov is with the Technical College by Plovdiv University "Paisiy Hilendarski", 4700 Smolyan, Bulgaria, e-mail: slubomirov@yahoo.com

The microprocessor is the basic unit of the digital relay protection module. Its input and output mechanisms ensure communication with the peripheral devices.

These additional modules make communication with output data transducers, control objects, operators, etc possible. Practically all information processing in relay protection is carried out by microprocessors according to a rigorously defined algorithm, realized as a computer program.

The specific characteristics of electricity transfer, distribution and consumption inevitably led to the development of automatic control technical mechanisms even at the early stages of their existence.

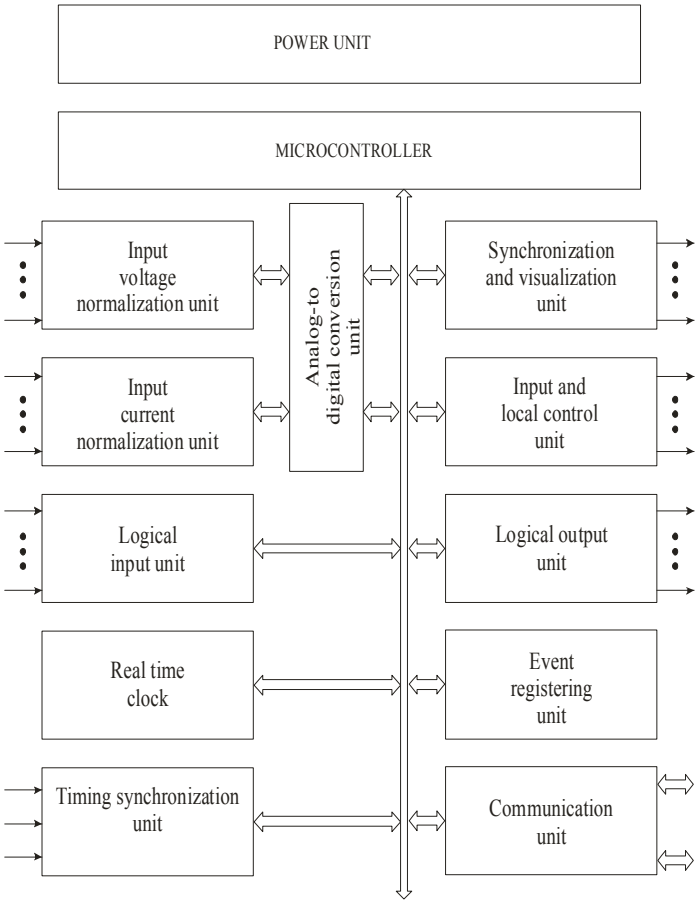


Figure 1. Digital protection relaying

II. APPROACHES FOR EXTERNAL COMMUNICATION PINS

For port 0 and port 1, the GPIO can be selected to be Fast GPIO or legacy GPIO. Port 2 and port 3 are available in the

144-pin packages only and are always legacy GPIO. Not all pins are available on port 0 and port 1. The respective bits in the GPIO registers are reserved.

• Every physical GPIO port can be accessed either through registers providing enhanced features and accelerated port access or through legacy registers providing backward compatibility to earlier LPC2000 devices.

• Accelerated Fast GPIO functions.

– GPIO registers are relocated to the ARM local bus so that the fastest possible I/O timing can be achieved.

– Mask registers allow treating sets of port bits as a group, leaving other bits unchanged.

– All registers are byte, half-word, and word addressable.

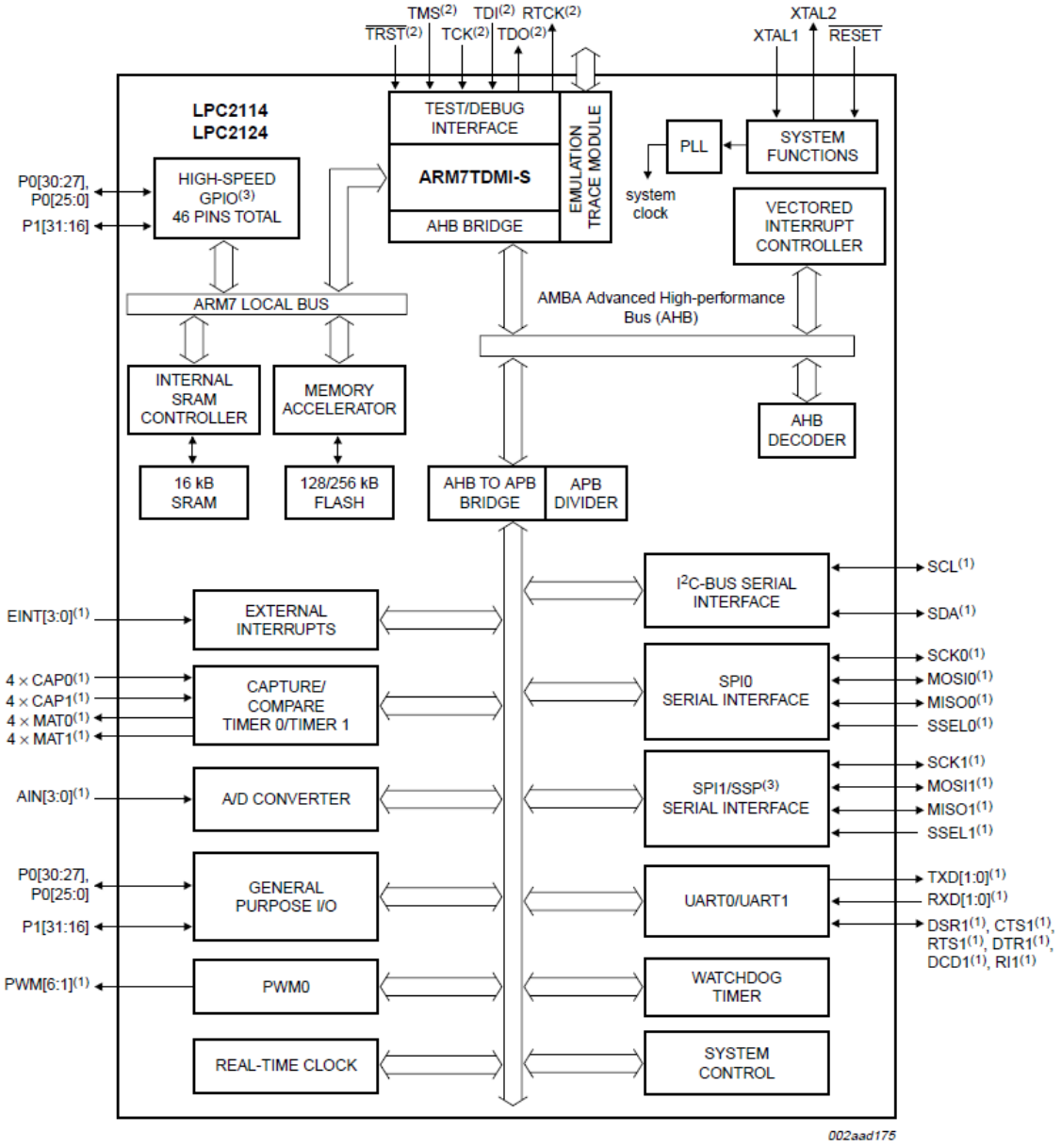
The entire port value can be written in one instruction.

• Bit-level set and clear registers allow a single instruction set or clear of any number of bits in one port.

• Direction of each pin can be controlled individually.

• All I/O default to inputs after reset.

III. Block Diagram



(1) Shared with GPIO.

(2) When test/debug interface is used, GPIO/other functions sharing these pins are not available.

(3) SSP interface and high-speed GPIO are available on LPC2114/01 and LPC2124/01 only.

IV. DESCRIPTION OF THE INPUT MODULE HARDWARE

LPC21xx/LPC22xx devices have two 32-bit General Purpose I/O ports. PORT0 and PORT1 are controlled by two groups of 4 registers. LPC22xx devices have two additional 32-bit ports, PORT2 and PORT3. These ports can be configured either as external memory data address and data bus or as GPIOs sharing pins with a handful of digital and analog functions. The functions and relative timing of older GPIO implementations is preserved.

The registers and enhanced Fast GPIO features available on the PORT0 and PORT1 only. All of these registers are located directly on the local bus of the CPU for the fastest possible read and write timing. An additional feature has been added that provides byte and half-word addressability of all GPIO registers. A mask register allows treating groups of bits in a single GPIO port separately from other bits on the same port.

When PORT0 and/or PORT1 are used, the user must select whether these ports will be accessed via registers that provide enhanced features or a legacy set of registers. While both of a port's fast and legacy GPIO registers are controlling the same physical pins, these two port control branches are mutually exclusive and operate independently. For example, changing a pin's output through a fast register will not be observable trough the corresponding legacy register.

The following text will refer to the legacy GPIO as "the slow" GPIO, while GPIO equipped

with the enhanced features will be referred as "the fast" GPIO. The "slow", legacy registers are word accessible only. The "fast" GPIO registers are byte,
half-word, and word accessible.

Example 1: sequential accesses to IOSET and IOCLR affecting the same GPIO pin/bit. The state of a GPIO pin configured as output is determined by writes into the pin's port IOSET and IOCLR registers. The last access to the IOSET/IOCLR register will determine the final output of the pin.
In the following code example

IO0DIR = 0x0000 0080 ;pin P0.7 configured as output
IO0CLR = 0x0000 0080 ;P0.7 goes LOW
IO0SET = 0x0000 0080 ;P0.7 goes HIGH
IO0CLR = 0x0000 0080 ;P0.7 goes LOW

pin P0.7 is configured as an output pin (write to IO0DIR register). Then, the P0.7 output pin is set to low (first write to IO0CLR register). A short high pulse follows on P0.7 (write access to IO0SET), and the second write to IO0CLR register sets pin P0.7 back to low level.

Example 2: an immediate output of 0s and 1s on a GPIO port Writing 1's to the port's IOSET register (setting port output to HIGH) followed by writing 1's to the IOCLR register (setting port output to LOW) causes a slight delay between the HIGH and LOW output at the port's pins.

There are systems that can tolerate this delay of a valid output, but for some applications simultaneous output of a binary content (mixed 0s and 1s) within a group of pins on a single GPIO port is required. This can be accomplished by writing to the port's IOPIN
register.

The following code will preserve existing output on PORT0 pins P0.[31:16] and P0.[7:0] and at the same time set P0.[15:8] to 0xA5, regardless of the previous value of pins P0.[15:8]:IO0PIN = (IO0PIN && 0xFFFF00FF) || 0x0000A500
The same outcome can be obtained using the fast port access.

Solution 1: using 32-bit (word) accessible fast GPIO registers

FIO0MASK = 0xFFFF00FF;
FIO0PIN = 0x0000A500;

Solution 2: using 16-bit (half-word) accessible fast GPIO registers

FIO0MASKL = 0x00FF;
FIO0PINL = 0xA500;

Solution 3: using 8-bit (byte) accessible fast GPIO registers

FIO0PIN1 = 0xA5;

Output signal frequency has to be considered when using the legacy and enhanced GPIO registers. The enhanced features of fast GPIO ports available on this microcontroller make the performance of the GPIO pins more dependent on the details of the application code. In particular, software access to a GPIO pin is 3.5 times faster through the fast GPIO registers than through the legacy set of registers. As a result, the maximum output frequency of the digital pin is increased 3.5 times if the fast GPIO registers are used. This tremendous increase of the output frequency is less noticeable when plain C code is used. The portion of an application handling the fast port output should be written in assembly code and executed in the ARM mode to take full advantage of the fast GPIO access.

The following is a code example in which the pin control section is written in assembly language for ARM. It illustrates the difference between the fast and slow GPIO port output capabilities. For the best performances, compile this code in the ARM mode and execute from the on-chip SRAM memory.

```
ldr r0,=0xe01fc1a0 /*register address--enable fast port*/
mov r1,#0x1
str r1,[r0] /*enable fast port0*/
ldr r1,=0xffffffff
ldr r0,=0x3fffc000 /*direction of fast port0*/
str r1,[r0]
ldr r0,=0xe0028018 /*direction of slow port 1*/
str r1,[r0]
ldr r0,=0x3fffc018 /*FIO0SET -- fast port0 register*/
ldr r1,=0x3fffc01c /*FIO0CLR0 -- fast port0 register*/
ldr r2,=0x00001000 /*select fast port 0.12 for toggle*/
ldr r3,=0xE0028014 /*IO1SET -- slow port1 register*/
ldr r4,=0xE002801C /*IO1CLR -- slow port1 register*/
ldr r5,=0x00100000 /*select slow port 1.20 for toggle*/
/*Generate 2 pulses on the fast port*/
str r2,[r0]
str r2,[r1]
str r2,[r0]
str r2,[r1]
```

```
/*Generate 2 pulses on the slow port*/
str r5,[r3]
str r5,[r4]
str r5,[r3]
str r5,[r4]
loop: b loop
```

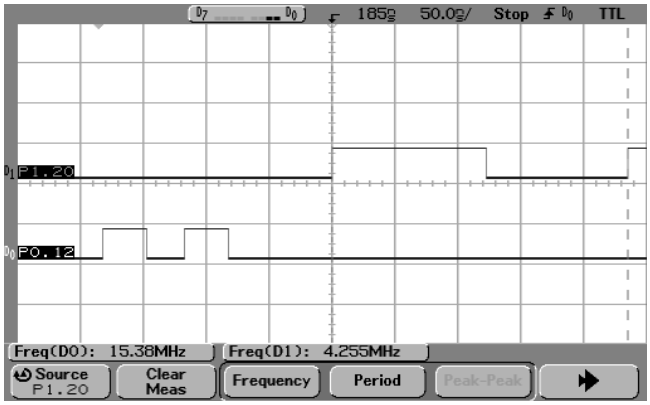
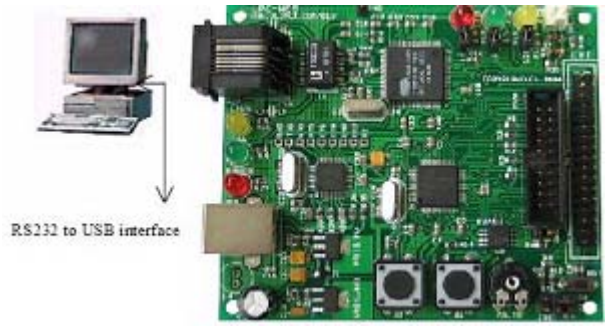


Fig.2. Illustration of the fast and slow GPIO access and output showing 3.5 x increase of the pin output frequency

VI. CONCLUSIONS

Apart from processing the measurements data the microcontroller carries out protection and control functions by performing the following actions:

- Filtering and processing of measurement signals.
- Constant monitoring of measurement values.
- Monitoring the conditions necessitating each protective function activation.
- Signal formation by transforming current and voltage input data in accordance with the algorithm of work for each protective function.
- Differential values and delay values formation.
- Phase current frequency and delay values analysis.
- Computation of absolute values of currents to determine the thermal characteristics and scanning of the temperature increase of the protected system.
- Monitoring the threshold values and timing synchronization.
- Logical functions signals processing.
- User defined logical functions processing.
- Decision making to issue a switch off command.
- Verification and issuing a command for the commutation devices.
- Storage of malfunction and failure signals information and the data necessary for systematic analysis of the breakdowns.
- Computation and visualization of measurement values and values arrived at after processing data
- Additional functions control, such as data storage, real time clock, communication, interfaces, etc.



The following advantages are clearly seen:

- Lower operational costs due to the self- diagnostic features.
- Capability for diagnostics not only of the relay protection, but also of the primary equipment.
- Less time necessary to investigate the reasons for the breakdown due to the registration and storage of malfunction mode information
- Automatic mode registration.
- Lower construction losses and lower losses of hardware.
- Accelerated switching off of short circuits.
- Unified technical solutions.
- Significant decrease of severe system breakdown risk.

ACKNOWLEDGEMENT

The investigations and the paper preparation have been supported by Technical University of Sofia within the framework of the project №091ис014-03/2009.

REFERENCES

[1] Ovcharov St., M.Bogdanov, An Investigation on a Transformer Converter of AC Current to AC Voltage, Proceedings of the Fifth National Applied Science Conference with International Participation "ELECTRONICS-ET '96", September 27-29, 1996, Sozopol, book 1, pp. 98-102

[2] Ovcharov St., M.Bogdanov, Investigation of the Amplitude and Phase Errors of Transformer Current to Voltage Converters, Proceedings of the Sixth National Applied Science Conference with International Participation "ELECTRONICS-ET '97", September 24-26, 1997, Sozopol, book 1, pp. 97-102

[3] Phadke, A.G. (1993) Synchronized phasor measurements in power systems. IEEE Comput. Appl. Power, 6 (2), 10–15.

[5] Phadke, A.G. and Thorp, J.S. (1988) Computer Relaying for Power Systems, Research Studies Press/John Wiley & Sons, Ltd.

http://www. www.olimex.com/dev

http://www.nxp.com