



Melec.ir



تقدیم به پدر و مادرم

که اولین و بزرگترین معلمانم که در مسیر پیشرفت، حامی و
همراهم بودند.

تقدیم به استادانی

که همچون شمع سوختن تا پله های ترقی را با دیدی بهتر
طی کنم.

حسینی و طاهونچی



فهرست

4	پیشگفتار
5	معرفی پروژه
5	• کاربری
5	• قطعات اصلی بکار گرفته شده در این پروژه
6	مقدمه ای بر تاریخچه ARM
8	• مشخصات کلی خانواده میکروکنترلر ARM
8	• کامپایلر استفاده شده جهت پروگرام ARM(NXP-LPC2138)
9	مشخصات میکروکنترلر NXP-LPC 2138
10	دیتا شیت پایه های NXP-LPC 2138
16	کاربرد وقفه ها در پروژه
17	ساختار وقفه و واحد VIC
19	• وقفه FIQ
19	• وقفه IRQ برداری
20	• وقفه IRQ غیر برداری
20	تایمر/کانتر با کاربرد عمومی
22	واحد UART
28	برنامه Source در کامپایلر Keil
33	برنامه LCD در کامپایلر Keil
35	مقدمه ای بر تاریخچه RFID
36	• TAG
36	• فرکانسها و استاندارد EPC
37	ماژول (EM-18)RFID و ارتباط آن با میکروکنترلر
38	مقدمه ای بر تاریخچه ماژول GSM(SIM 900)
43	ماژول GSM(SIM 900)
44	منابع و تشکر



پیشگفتار

با پیشرفت چشمگیر علم امروزه دفاتر حضور و غیاب قدیمی که دارای جداول بسیاری بودند و حتما نیاز به یک اپراتور جهت ثبت اطلاعات داشتند جای خود را به سیستم های حضور و غیاب مغناطیس و RFID داده اند به طوری که در بیشتر ادارات کارمندان پس از حضور در محل کار با قرار دادن کارت خود بر روی دستگاه کارت خوان ساعت حضور و خروج خود را ثبت می کنند. دستگاه های کارت خوان وسائط حمل و نقل عمومی نیز همچون مترو و اتوبوس که باعث تسریع در امور حمل و نقل و کاهش هزینه در خرید بلیت های کاغذی واز بین رفتن مقادیر زیادی کاغذ از جمله ی آنهاست .

با وجود این همه سیستم های پیشرفته هنوز در مدارس ما به جزء بعضی از مدارس غیر دولتی سیستم حضور و غیاب به صورت سنتی بوده ؛ به این صورت که هر معلم پس از حضور در سر کلاس باید اقدام به حضور و غیاب می کرد و اگر دانش آموز غیبت داشت بعد از 3 بار غیبت والدین او را در جریان قرار می دادند. از طرفی بعضی از والدین از حضور یا عدم حضور فرزندشان در مدرسه بی خبر بودند و همین موضوع باعث دلنگرانی آنها می شد.

در کارگاه ها و یا کارخانه های کوچک نیز رئیس مربوطه تنها در زمان پرداخت حقوق از وضعیت حضور و غیاب کارکنان خود آگاه می شد.

اما در این پروژه سعی شد که این قضایا بر طرف شود و والدین ویا رئیس کارخانه یا کارگاه به صورت آنلاین از حضور دانش آموز یا کارکنان خود در مدرسه یا محل کار آگاه شوند که در ادامه پایان نامه به آن پرداخته شده است.

در ضمن قسمت های که نیاز به دیتاشیت داشت مستقیماً در همان قسمت آورده شده لذا از آوردن کامل دیتاشیت LPC 2138 در آخر پایان نامه خودداری شده است.



معرفی پروژه :

سیستم حضور غیاب از طریق کارت RF ID و ارسال اطلاعات به صورت SMS با میکروکنترلر ARM.

در این پروژه نحوه ی عملکرد به این صورت است که هر کاربر دارای یک کارت RF ID می باشد (که درون آن یک کد که حاوی اطلاعات کاربر می باشد تعریف شده است) و وقتی کاربر در محل کار حضور پیدا می کند و کارت را جلوی RF ID می گیرد سیستم ورود شخص را ثبت کرده و با ارسال SMS اگر کاربر دانش آموز باشد به والدین آن و اگر کارکنان کارخانه باشند به رئیس مربوطه زمان حضور آنها را اعلام می کند.

البته این پروژه قابلیت ارسال SMS برای چند نفر را هم داراست اما برای کاهش هزینه این عمل برای یک نفر صورت می گیرد.

در این پروژه option های دیگری چون LCD اضافه شد که شخص ثبت حضور خود را در محل رویت کند با ثبت ساعت حضور .

• کاربری

میتوان از این سیستم در مدارس استفاده کرد و مدیر مدرسه و والدین دانش آموز را از حضور دانش آموز در مدرسه مطلع ساخت ; یا در ادارات و شرکتها .

• قطعات اصلی بکار گرفته شده در این پروژه

در این پروژه سعی میشود از میکروکنترلر ARM (NXP-LPC2138) استفاده شود و زبان برنامه نویسی C و ماژول GSM SIM 900 و ماژول RFID (EM-18) و IC RS232.



مقدمه ای بر تاریخچه ARM :

در اوایل دهه‌ی 80 اولین پردازنده‌ی ARM در شرکت ACRON طراحی و به بازار ارائه گردید. تقریباً 80 سال بعد و در سال 1990 با پیوستن شرکت‌های APPLE و VLST Technology به کمپانی ACRON شرکت ARM تأسیس شد. این شرکت با طراحی معماری‌های مختلف از پردازنده‌های RISC و ارائه‌ی آن به بازار توانست در مدت کوتاهی 75 درصد بازار پردازنده‌های 32 بیتی را در دست بگیرد.

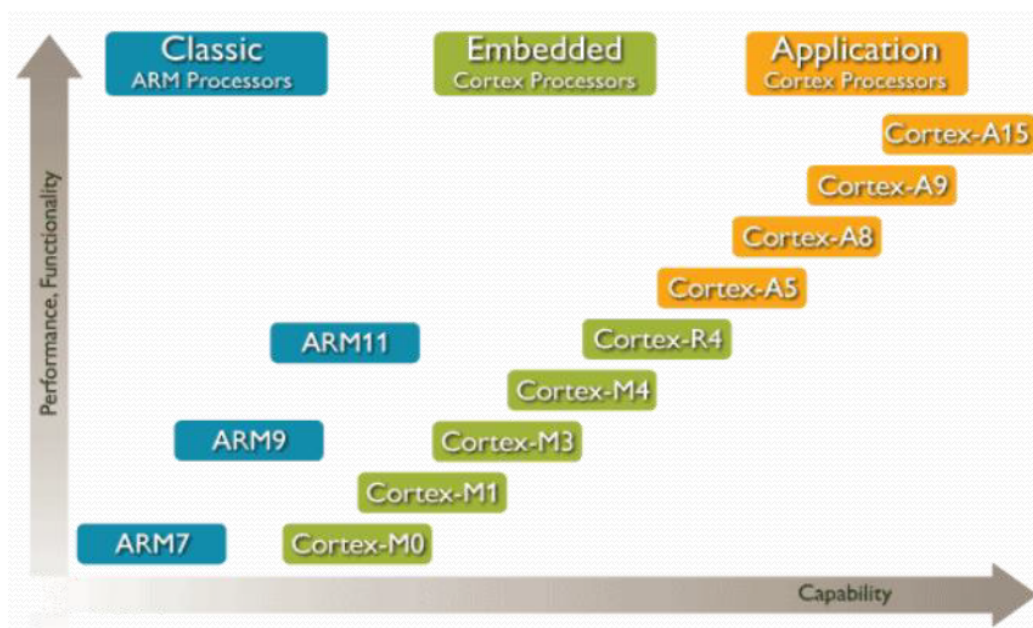
امروزه بیش از 90 شرکت معماری طراحی شده توسط شرکت ARM را خریداری کرده و بعد از افزودن امکانات جانبی، آن را به بازار عرضه می‌کنند.

در واقع ARM فقط یک معماری (با یک نقشه) می‌باشد که به شرکت‌های تولیدکننده‌ی میکروکنترلر فروخته می‌شود و آنها باید تحت لیسانس شرکت ARM آن را تولید نمایند. وجود معماری متناسب و قابلیت توسعه‌ی سخت افزار به صورت نامحدود و ... از ویژگی‌های این محصول می‌باشد.





هسته های که تاکنون ارائه شده اند ARM7 ، ARM9 ، ARM11 و ... می باشند. همچنین از این هسته ها زیرمجموعه های همچون Cortex – M0 و Cortex-M3 بوجود آمده است.



همانطور که در شکل فوق مشخص شده است بلوک های آبی (ستون اول) ARM v4-6 و بلوک های سبز (ستون دوم) ARM v7-M تا ARM v7-R و بلوک های نارنجی (ستون سوم) ARM v7-A

به دلیل یکسان بودن هسته در تمامی میکروکنترلرها (محصولات شرکت های مختلف) کار با تمامی آنها تقریباً یکسان است و شما می توانید با خیال راحت با هسته ی دلخواه کار کنید.

حوزه های کاربرد

Application Processors	Embedded Processors	SecurCores
ARM Cortex-A5	ARM Cortex-M0	SC300
ARM Cortex-A8	ARM Cortex-M1	SecurCore SC100
ARM Cortex-A9 MPCore	ARM Cortex-M3	SecurCore SC200
ARM Cortex-A9 Single Core Processor	ARM Cortex-R4(F)	
ARM11 MPCore	ARM1156T2(F)-S	
ARM1136J(F)-S	ARM7EJ-S	
ARM1176JZ(F)-S	ARM7TDMI	
ARM720T	ARM7TDMI-S	
ARM920T	ARM946E-S	
ARM922T	ARM966E-S	
ARM926EJ-S	ARM968E-S	



امروزه بیش از 90 شرکت میکروکنترلرهای مبتنی بر هسته ی ARM را تولید می کنند که حاصل این تولیدات بیش از 4 هزار میکرو کنترلر است. در این بین شرکت های نرم افزاری نیز فعالیت گسترده ای دارند، در دنیای نرم افزار بیش از 50 کامپایلر برای میکروکنترلرهای که از هسته ی ARM استفاده می کنند، وجود دارد که می توانیم کامپایلرهای معروف KEIL و IAR را در این زمینه معرفی نماییم.

مشخصات کلی خانواده میکروکنترلر ARM :

- دارای بیش از 200 زیرمجموعه
 - دارای حداکثر فرکانس کاری بیش از 1GHz که حداکثر کلاک ذکر شده حاصل Over Clock پردازنده می باشد (با تنظیمات واحد PLL)
 - قدرت پردازش عمومی و اختصاصی بالایی دارد که منظور از قدرت پردازش عمومی و اختصاصی، سرعت و قدرت پردازش اطلاعات در مصارف عمومی (مانند کارهای کنترولی) و اختصاصی (مانند پردازش تصویر) می باشد.
 - نویز پذیری بسیار کم
- پشتیبانی از پروتکل های ارتباطی: پروتکل های که عموماً در ایران استفاده می شود. عبارتند از I2C، Can، RS232، USB و ... می باشد.

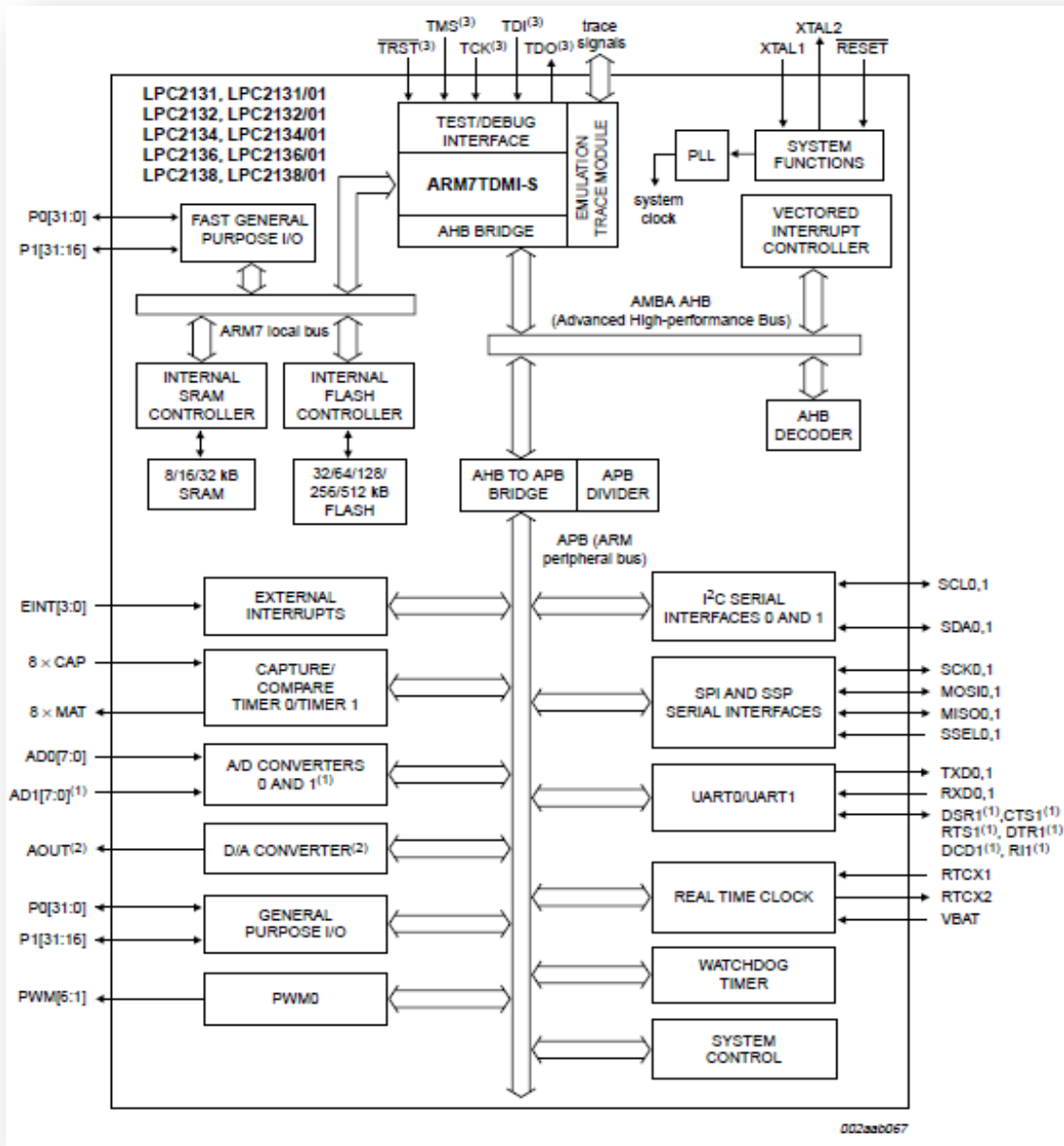
کامپایلر استفاده شده جهت پروگرام ARM(NXP-LPC2138)

Keil ARM که قابلیت برنامه نویسی میکروکنترلرهای ARM به زبان های C و ++C و اسمبلی / امکان شبیه سازی برنامه نوشته شده / پشتیبانی از تمامی میکروکنترلرهای ARM / دارای منابع آموزشی متوسط / محیط حرفه ای و استفاده از ویرایشگر قوی / یادگیری سریع نرم افزار.





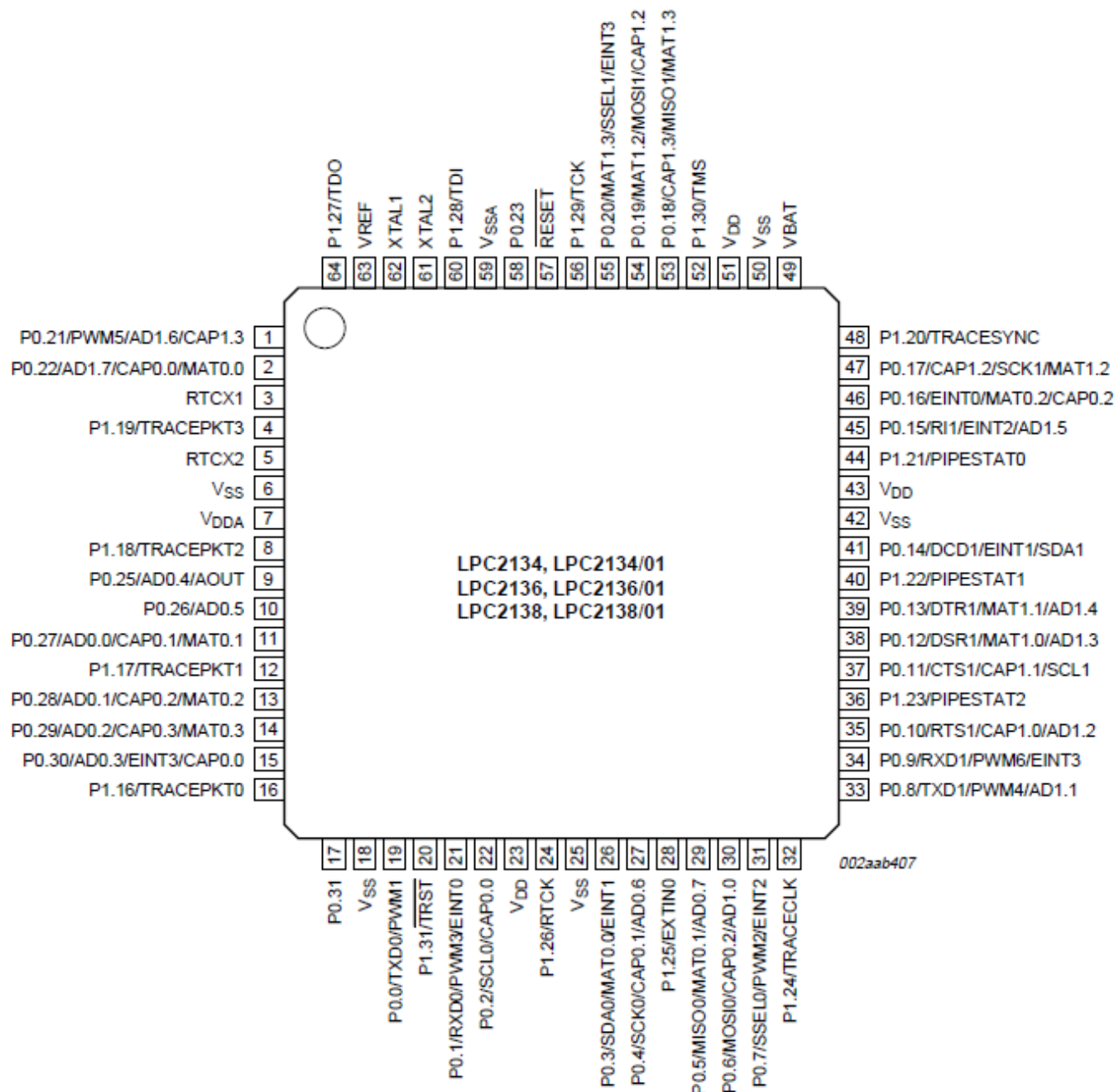
مشخصات میکروکنترلر NXP-LPC 2138 :



در این پروژه از 3 وقفه TIMERO و UART0 و UART1 استفاده شده است که در ادامه این وقفه ها توضیح داده شده اند.



دیتا شیت پایه های NXP-LPC 2138



برای بایاسینگ ARM این پروژه باید حتماً پایه های 63 و 51 و 49 و 43 و 23 و 7 (Vcc) شوند و پایه های 59 و 50 و 42 و 25 و 18 و 6 (GND) شوند و پایه های 57 و 48 و 41 و 24 و 20 و 17 (Pull up با مقاومت $1\text{ k}\Omega$) شوند در ضمن جهت نویزگیری ARM دور تا دور IC از خازن های سرامیکی 100 nF استفاده شده است.



Pin description

Symbol	Pin	Type	Description
P0.0 to P0.31		I/O	Port 0: Port 0 is a 32-bit I/O port with individual direction controls for each bit. Total of 30 pins of the Port 0 can be used as a general purpose bi-directional digital I/Os while P0.31 is output only pin. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pin P0.24 is not available.
P0.0/TXD0/ PWM1	19 ^[1]	I/O	P0.0 — General purpose digital input/output pin.
		O	TXD0 — Transmitter output for UART0.
		O	PWM1 — Pulse Width Modulator output 1.
P0.1/RxD0/ PWM3/EINT0	21 ^[2]	I/O	P0.1 — General purpose digital input/output pin.
		I	RxD0 — Receiver input for UART0.
		O	PWM3 — Pulse Width Modulator output 3.
		I	EINT0 — External interrupt 0 input.
P0.2/SCL0/ CAP0.0	22 ^[3]	I/O	P0.2 — General purpose digital input/output pin.
		I/O	SCL0 — I ² C0 clock input/output. Open drain output (for I ² C compliance).
		I	CAP0.0 — Capture input for Timer 0, channel 0.
P0.3/SDA0/ MAT0.0/EINT1	26 ^[3]	I/O	P0.3 — General purpose digital input/output pin.
		I/O	SDA0 — I ² C0 data input/output. Open drain output (for I ² C compliance).
		O	MAT0.0 — Match output for Timer 0, channel 0.
		I	EINT1 — External interrupt 1 input.
P0.4/SCK0/ CAP0.1/AD0.6	27 ^[4]	I/O	P0.4 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SPI0. SPI clock output from master or input to slave.
		I	CAP0.1 — Capture input for Timer 0, channel 0.
		I	AD0.6 — A/D converter 0, input 6.
P0.5/MISO0/ MAT0.1/AD0.7	29 ^[4]	I/O	P0.5 — General purpose digital input/output pin.
		I/O	MISO0 — Master In Slave Out for SPI0. Data input to SPI master or data output from SPI slave.
		O	MAT0.1 — Match output for Timer 0, channel 1.
		I	AD0.7 — A/D converter 0, input 7.
P0.6/MOSI0/ CAP0.2/AD1.0	30 ^[4]	I/O	P0.6 — General purpose digital input/output pin.
		I/O	MOSI0 — Master Out Slave In for SPI0. Data output from SPI master or data input to SPI slave.
		I	CAP0.2 — Capture input for Timer 0, channel 2
		I	AD1.0 — A/D converter 1, input 0 (LPC2134/6/8 and matching /01 only).
P0.7/SSEL0/ PWM2/EINT2	31 ^[2]	I/O	P0.7 — General purpose digital input/output pin.
		I	SSEL0 — Slave Select for SPI0. Selects the SPI interface as a slave.
		O	PWM2 — Pulse Width Modulator output 2.
		I	EINT2 — External interrupt 2 input.
P0.8/TXD1/ PWM4/AD1.1	33 ^[4]	I/O	P0.8 — General purpose digital input/output pin.
		O	TXD1 — Transmitter output for UART1.
		O	PWM4 — Pulse Width Modulator output 4.
		I	AD1.1 — A/D converter 1, input 1 (LPC2134/6/8 and matching /01 only).



Pin description ...continued

Symbol	Pin	Type	Description
P0.9/RxD1/ PWM6/EINT3	34 ^[2]	I/O	P0.9 — General purpose digital input/output pin.
		I	RxD1 — Receiver input for UART1.
		O	PWM6 — Pulse Width Modulator output 6.
		I	EINT3 — External interrupt 3 input.
P0.10/RTS1/ CAP1.0/AD1.2	35 ^[4]	I/O	P0.10 — General purpose digital input/output pin.
		O	RTS1 — Request to Send output for UART1; available in LPC2134/6/8 and matching /01 devices only.
		I	CAP1.0 — Capture input for Timer 1, channel 0.
		I	AD1.2 — A/D converter 1, input 2 (LPC2134/6/8 and matching /01 only).
P0.11/CTS1/ CAP1.1/SCL1	37 ^[3]	I/O	P0.11 — General purpose digital input/output pin.
		I	CTS1 — Clear to Send input for UART1; available in LPC2134/6/8 and matching /01 devices only
		I	CAP1.1 — Capture input for Timer 1, channel 1.
		I/O	SCL1 — I ² C1 clock input/output. Open drain output (for I ² C compliance)
P0.12/DSR1/ MAT1.0/AD1.3	38 ^[4]	I/O	P0.12 — General purpose digital input/output pin
		I	DSR1 — Data Set Ready input for UART1. Available in LPC2134/6/8 only.
		O	MAT1.0 — Match output for Timer 1, channel 0.
		I	AD1.3 — A/D converter input 3 (LPC2134/6/8 and matching /01 only).
P0.13/DTR1/ MAT1.1/AD1.4	39 ^[4]	I/O	P0.13 — General purpose digital input/output pin
		O	DTR1 — Data Terminal Ready output for UART1; available in LPC2134/6/8 and matching /01 devices only.
		O	MAT1.1 — Match output for Timer 1, channel 1.
		I	AD1.4 — A/D converter input 4 (LPC2134/6/8 and matching /01 only).
P0.14/DCD1/ EINT1/SDA1	41 ^[3]	I/O	P0.14 — General purpose digital input/output pin.
		I	DCD1 — Data Carrier Detect input for UART1; available in LPC2134/6/8 and matching /01 devices only.
		I	EINT1 — External interrupt 1 input.
		I/O	SDA1 — I ² C1 data input/output. Open drain output (for I ² C compliance) Note: LOW on this pin while RESET is LOW forces on-chip boot-loader to take over control of the part after reset.
P0.15/RI1/ EINT2/AD1.5	45 ^[4]	I/O	P0.15 — General purpose digital input/output pin.
		I	RI1 — Ring Indicator input for UART1; available in LPC2134/6/8 and matching /01 devices only.
		I	EINT2 — External interrupt 2 input.
		I	AD1.5 — A/D converter 1, input 5 (LPC2134/6/8 and matching /01 only).
P0.16/EINT0/ MAT0.2/CAP0.2	46 ^[2]	I/O	P0.16 — General purpose digital input/output pin.
		I	EINT0 — External interrupt 0 input.
		O	MAT0.2 — Match output for Timer 0, channel 2.
		I	CAP0.2 — Capture input for Timer 0, channel 2.



Pin description ...continued

Symbol	Pin	Type	Description
P0.17/CAP1.2/ SCK1/MAT1.2	47 ^[1]	I/O	P0.17 — General purpose digital input/output pin.
		I	CAP1.2 — Capture input for Timer 1, channel 2.
		I/O	SCK1 — Serial Clock for SSP. Clock output from master or input to slave.
		O	MAT1.2 — Match output for Timer 1, channel 2.
P0.18/CAP1.3/ MISO1/MAT1.3	53 ^[1]	I/O	P0.18 — General purpose digital input/output pin.
		I	CAP1.3 — Capture input for Timer 1, channel 3.
		I/O	MISO1 — Master In Slave Out for SSP. Data input to SPI master or data output from SSP slave.
P0.19/MAT1.2/ MOSI1/CAP1.2	54 ^[1]	O	MAT1.2 — Match output for Timer 1, channel 2.
		I/O	MOSI1 — Master Out Slave In for SSP. Data output from SSP master or data input to SSP slave.
		I	CAP1.2 — Capture input for Timer 1, channel 2.
P0.20/MAT1.3/ SSEL1/EINT3	55 ^[2]	I/O	P0.20 — General purpose digital input/output pin.
		O	MAT1.3 — Match output for Timer 1, channel 3.
		I	SSEL1 — Slave Select for SSP. Selects the SSP interface as a slave.
		I	EINT3 — External interrupt 3 input.
P0.21/PWM5/ AD1.6/CAP1.3	1 ^[4]	I/O	P0.21 — General purpose digital input/output pin.
		O	PWM5 — Pulse Width Modulator output 5.
		I	AD1.6 — A/D converter 1, input 6 (LPC2134/6/8 and matching /01 only).
		I	CAP1.3 — Capture input for Timer 1, channel 3.
P0.22/AD1.7/ CAP0.0/MAT0.0	2 ^[4]	I/O	P0.22 — General purpose digital input/output pin.
		I	AD1.7 — A/D converter 1, input 7 (LPC2134/6/8 and matching /01 only).
		I	CAP0.0 — Capture input for Timer 0, channel 0.
		O	MAT0.0 — Match output for Timer 0, channel 0.
P0.23	58 ^[1]	I/O	P0.23 — General purpose digital input/output pin.
P0.25/AD0.4/ Aout	9 ^[5]	I/O	P0.25 — General purpose digital input/output pin.
		O	Aout — D/A converter output; available in LPC2132/4/6/8 and matching /01 devices only.
P0.26/AD0.5	10 ^[4]	I/O	P0.26 — General purpose digital input/output pin.
		I	AD0.5 — A/D converter 0, input 5.
P0.27/AD0.0/ CAP0.1/MAT0.1	11 ^[4]	I/O	P0.27 — General purpose digital input/output pin.
		I	AD0.0 — A/D converter 0, input 0.
		O	MAT0.1 — Match output for Timer 0, channel 1.
P0.28/AD0.1/ CAP0.2/MAT0.2	13 ^[4]	I/O	P0.28 — General purpose digital input/output pin.
		I	AD0.1 — A/D converter 0, input 1.
		O	MAT0.2 — Match output for Timer 0, channel 2.



Pin description ...continued

Symbol	Pin	Type	Description
P0.29/AD0.2/ CAP0.3/MAT0.3	14 ⁽⁴⁾	I/O	P0.29 — General purpose digital input/output pin.
		I	AD0.2 — A/D converter 0, input 2.
		I	CAP0.3 — Capture input for Timer 0, Channel 3.
		O	MAT0.3 — Match output for Timer 0, channel 3.
P0.30/AD0.3/ EINT3/CAP0.0	15 ⁽⁴⁾	I/O	P0.30 — General purpose digital input/output pin.
		I	AD0.3 — A/D converter 0, input 3.
		I	EINT3 — External interrupt 3 input.
P0.31	17 ⁽⁶⁾	O	P0.31 — General purpose digital output only pin Note: This pin MUST NOT be externally pulled LOW when RESET pin is LOW or the JTAG port will be disabled.
		I/O	Port 1: Port 1 is a 32-bit bi-directional I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 0 through 15 of port 1 are not available.
P1.16/ TRACEPKT0	16 ⁽⁶⁾	I/O	P1.16 — General purpose digital input/output pin.
		O	TRACEPKT0 — Trace Packet, bit 0. Standard I/O port with internal pull-up.
P1.17/ TRACEPKT1	12 ⁽⁶⁾	I/O	P1.17 — General purpose digital input/output pin.
		O	TRACEPKT1 — Trace Packet, bit 1. Standard I/O port with internal pull-up.
P1.18/ TRACEPKT2	8 ⁽⁶⁾	I/O	P1.18 — General purpose digital input/output pin.
		O	TRACEPKT2 — Trace Packet, bit 2. Standard I/O port with internal pull-up.
P1.19/ TRACEPKT3	4 ⁽⁶⁾	I/O	P1.19 — General purpose digital input/output pin.
		O	TRACEPKT3 — Trace Packet, bit 3. Standard I/O port with internal pull-up.
P1.20/ TRACESYNC	48 ⁽⁶⁾	I/O	P1.20 — General purpose digital input/output pin.
		O	TRACESYNC — Trace Synchronization. Standard I/O port with internal pull-up. Note: LOW on this pin while $\overline{\text{RESET}}$ is LOW enables pins P1.25:16 to operate as Trace port after reset
P1.21/ PIPESTAT0	44 ⁽⁶⁾	I/O	P1.21 — General purpose digital input/output pin.
		O	PIPESTAT0 — Pipeline Status, bit 0. Standard I/O port with internal pull-up.
P1.22/ PIPESTAT1	40 ⁽⁶⁾	I/O	P1.22 — General purpose digital input/output pin.
		O	PIPESTAT1 — Pipeline Status, bit 1. Standard I/O port with internal pull-up.
P1.23/ PIPESTAT2	38 ⁽⁶⁾	I/O	P1.23 — General purpose digital input/output pin.
		O	PIPESTAT2 — Pipeline Status, bit 2. Standard I/O port with internal pull-up.
P1.24/ TRACECLK	32 ⁽⁶⁾	I/O	P1.24 — General purpose digital input/output pin.
		O	TRACECLK — Trace Clock. Standard I/O port with internal pull-up.
P1.25/EXTIN0	28 ⁽⁶⁾	I/O	P1.25 — General purpose digital input/output pin.
		I	EXTIN0 — External Trigger Input. Standard I/O with internal pull-up.



Pin description ...continued

Symbol	Pin	Type	Description
P1.26/RTCK	24 ^[6]	I/O	P1.26 — General purpose digital input/output pin.
		I/O	RTCK — Returned Test Clock output. Extra signal added to the JTAG port. Assists debugger synchronization when processor frequency varies. Bi-directional pin with internal pull-up. Note: LOW on this pin while $\overline{\text{RESET}}$ is LOW enables pins P1.31:26 to operate as Debug port after reset
P1.27/TDO	64 ^[6]	I/O	P1.27 — General purpose digital input/output pin.
		O	TDO — Test Data out for JTAG interface.
P1.28/TDI	60 ^[6]	I/O	P1.28 — General purpose digital input/output pin.
		I	TDI — Test Data in for JTAG interface.
P1.29/TCK	56 ^[6]	I/O	P1.29 — General purpose digital input/output pin.
		I	TCK — Test Clock for JTAG interface.
P1.30/TMS	52 ^[6]	I/O	P1.30 — General purpose digital input/output pin.
		I	TMS — Test Mode Select for JTAG interface.
P1.31/ $\overline{\text{TRST}}$	20 ^[6]	I/O	P1.31 — General purpose digital input/output pin.
		I	$\overline{\text{TRST}}$ — Test Reset for JTAG interface.
$\overline{\text{RESET}}$	57 ^[7]	I	External reset input: A LOW on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.
XTAL1	62 ^[8]	I	Input to the oscillator circuit and internal clock generator circuits.
XTAL2	61 ^[8]	O	Output from the oscillator amplifier.
RTCX1	3 ^[8]	I	Input to the RTC oscillator circuit.
RTCX2	5 ^[8]	O	Output from the RTC oscillator circuit.
V _{SS}	6, 18, 25, 42, 50	I	Ground: 0 V reference.
V _{SSA}	59	I	Analog Ground: 0 V reference. This should nominally be the same voltage as V _{SS} , but should be isolated to minimize noise and error.
V _{DD}	23, 43, 51	I	3.3 V Power Supply: This is the power supply voltage for the core and I/O ports.
V _{DDA}	7	I	Analog 3.3 V Power Supply: This should be nominally the same voltage as V _{DD} but should be isolated to minimize noise and error. This voltage is used to power the ADC(s) and DAC (where available).
V _{REF}	63	I	A/D Converter Reference: This should be nominally the same voltage as V _{DD} but should be isolated to minimize noise and error. Level on this pin is used as a reference for A/D converter and DAC (where available).
V _{BAT}	49	I	RTC Power Supply: 3.3 V on this pin supplies the power to the RTC.

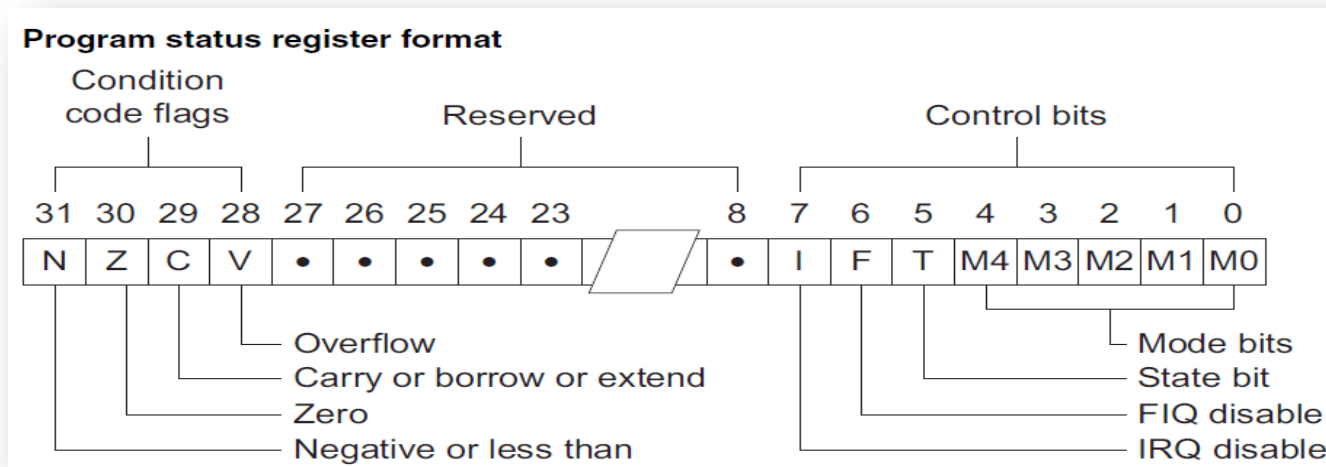
- [1] Bidirectional pin; Plain Input; 3 State Output; 10 ns Slew rate Control; TTL with Hysteresis; 5 V Tolerant.
- [2] Bidirectional; Input Glitch Filter (pulses shorter than 4 ns are ignored); 3 State Output; 10 ns Slew rate Control; TTL with Hysteresis.
- [3] I²C Pad; 400 kHz Specification; Open Drain; 5 V Tolerant.
- [4] Bidirectional; Input Glitch Filter (pulses shorter than 4 ns are ignored); Analog I/O; digital receiver disable; 3 State Output; 10 ns Slew Rate Control; TTL with Hysteresis; 5 V Tolerant
- [5] Bidirectional; Analog I/O; digital receiver disable; 3 State Output; 10 ns Slew Rate Control; TTL with Hysteresis; DAC enable output.
- [6] Bidirectional pin; Plain Input; 3 State Output; 10 ns Slew rate Control; TTL with Hysteresis; Pull-up; 5 V Tolerant.
- [7] Input; TTL with Hysteresis; 5 V Tolerant (pulses shorter than 20 ns are ignored).
- [8] Analog like pads having ESD structures only.



کاربرد وقفه ها در پروژه:

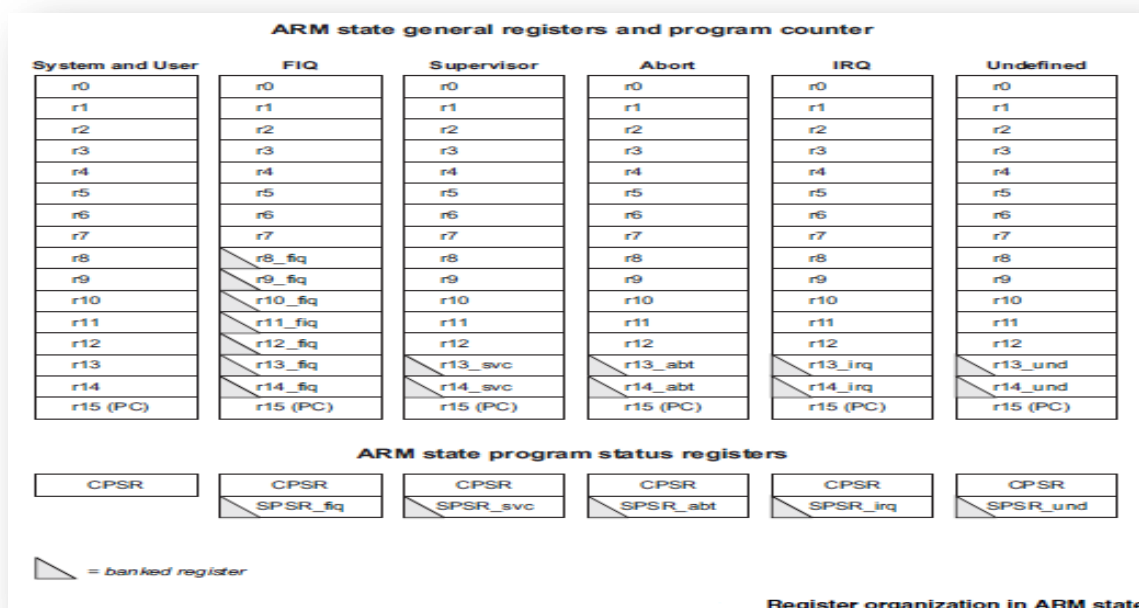
در معماری ARM وقفه ها حالت خاصی از اکسپشن ها (Exception) هستند که می توانند جریان اجرای برنامه را متوقف کرده و به بخش دیگر از حافظه منشعب شوند.

پردازنده ARM دارای 2 بیت فعال ساز وقفه های IRQ و FIQ در رجیستر CPSR است که می توان هر یک از این 2 وقفه را فعال و یا غیر فعال کرد.



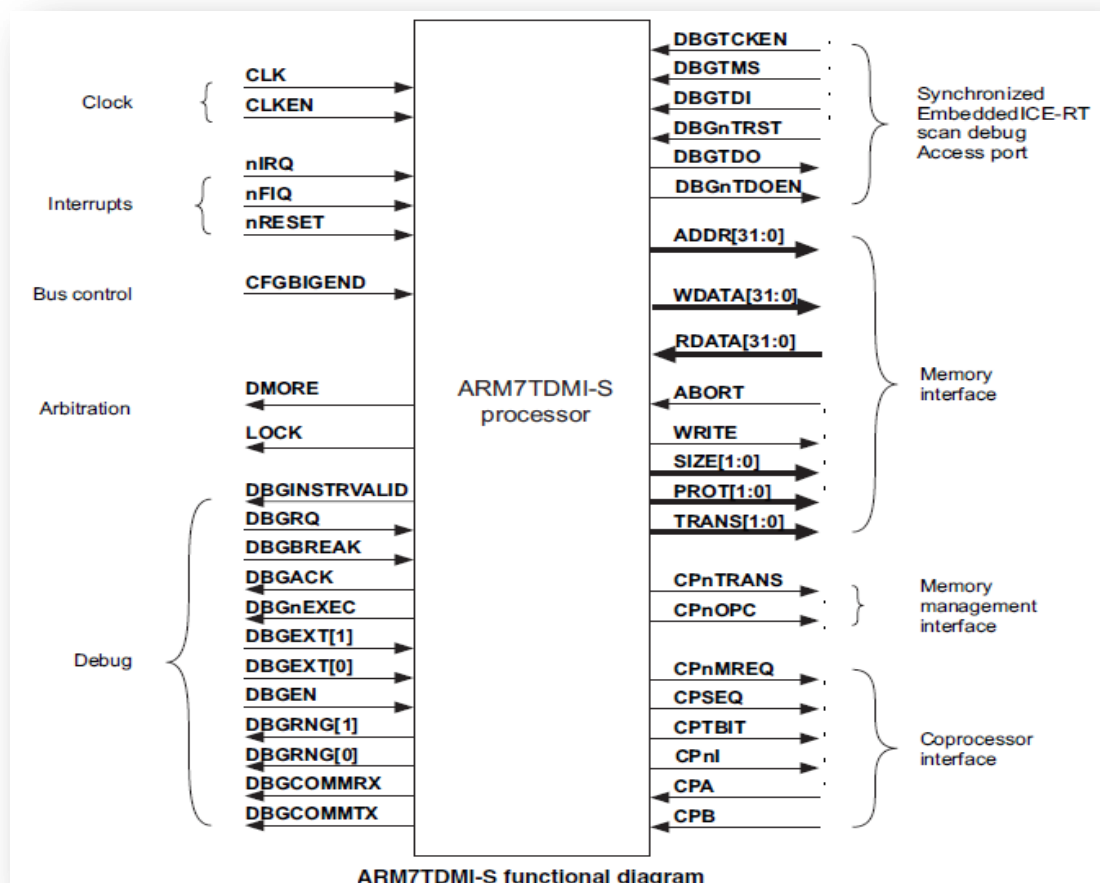
این ویژگی فعال و یا غیر فعال کردن وقفه ها در رجیستر CPSR یکی از مهمترین تفاوت های وقفه ها با سایر اکسپشن ها است چرا که IRQ و FIQ تنها وقفه هایی هستند که قابل فعال و یا غیر فعال شدن می باشند.

سایر اکسپشن ها مانند Reset ، Undefined Instruction ، Memory abort همواره فعال هستند.



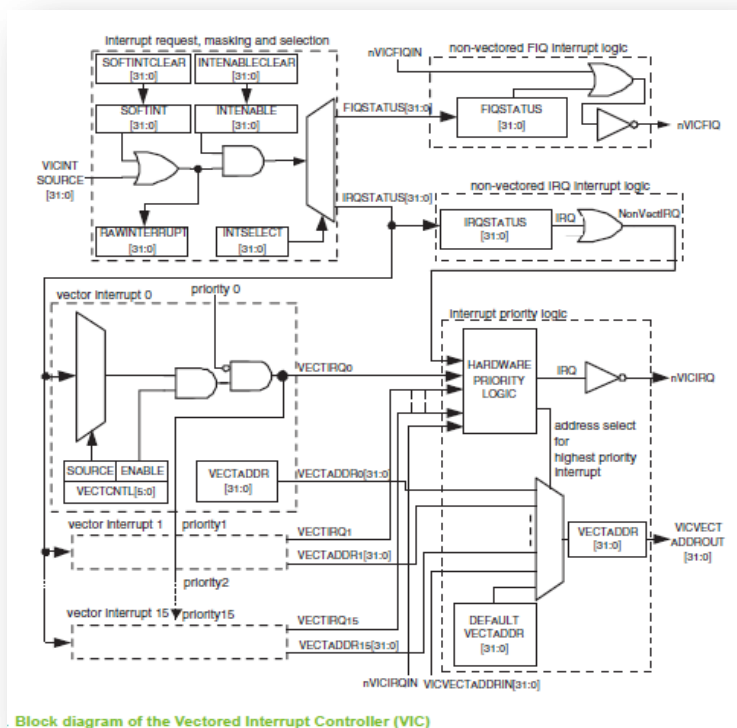


در این قسمت نگاهی خواهیم داشت به وقفه هایی با کاربرد عمومی یا IRQ (general Interrupt ReQuest) و وقفه سریع و یا FIQ (Fast Interrupt ReQuest). این 2 وقفه ، سیگنال های ورودی به نام nFIQ و nIRQ از پردازنده ARM هستند. که تمام رویدادهای بیرون از CPU توسط آن ها رسیدگی می شود.

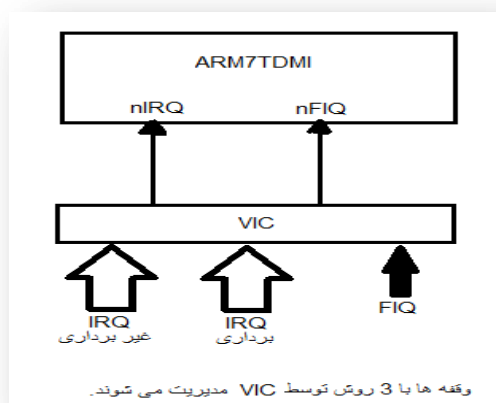


ساختار وقفه و واحد VIC:

پردازنده ARM7 دارای 2 سیگنال وقفه به نام های nFIQ و nIRQ است. به عنوان یک قانون کلی در سیستم های مبتنی بر ARM7 لازم است تنها یک منبع وقفه به nFIQ وصل شود تا پردازنده وارد این مد شده و با بالاترین سرعت ممکن شروع به پردازش وقفه کند. این بدان معنا است که تمام منابع دیگر وقفه باید به nIRQ متصل شوند. در یک سیستم ساده این کار می تواند از طریق یک گیت OR انجام شود.



به این صورت که وقتی سیگنال nIRQ فعال می شود CPU مجبور است تمام سخت افزار های جنبی را بررسی کند تا منع وقفه را شناسایی کند از آن جایی که این روش کند است و امکان اولویت دهی سخت افزاری نیز وجود ندارد به روش مناسب تری نیاز است. بدین منظور واحدی به نام "کنترل کننده وقفه برداری" یا VIC (Vectored Interrupt Controller) به میکرو کنترلر افزوده شده است. واحد VIC وظیفه مدیریت و رسیدگی به تمام وقفه های در خواست شده از سخت افزار های جنبی را بر عهده دارد. در قطعات سری LPC2000 دارای 32 ورودی است و هر یک از این منابع وقفه می توانند یکی از ورودی های آن متصل شوند. برنامه نوشته شده می تواند هر یک از این ورودی ها را به یکی از سیگنال های وقفه CPU (خط nFIQ یا nIRQ) به 3 طریق وصل کند.





وقفه FIQ ، وقفه برداری و وقفه IRQ غیر برداری. زمان پاسخ گویی به این وقفه در این 3 روش متفاوت است؛ FIQ سریع ترین شیوه رسیدگی به وقفه هاست ، پس از آن برداری قرار دارد و در نهایت IRQ غیر برداری که کند ترین روش است.

وقفه FIQ :

هر یک از منابع وقفه می توانند به عنوان FIQ انتخاب شوند. رجیستر انتخاب وقفه VICIntSelect (VIC Interrupt Select) برای هر منبع وقفه یک بیت دارد که با ست کردن آن وقفه مورد نظر به عنوان FIQ در نظر گرفته می شود. اگر چه امکان ست کردن چند بیت از رجیستر VICIntSelect وجود دارد اما ایده آل آن است که تنها یک منبع وقفه FIQ داشته باشیم. پس از این که منبع وقفه FIQ را انتخاب کردید می توانید از طریق رجیستر فعال ساز وقفه یا VICIntEnable (VIC Interrupt Enable) آن را فعال کنید.

همانطور که رجیسترهای واحد VIC را تنظیم می کنید لازم است سخت افزار جنبی مربوط به آن را نیز تنظیم کرده و وقفه آن را فعال کنید. با فرض اینکه بیت فعال ساز وقفه FIQ در رجیستر CPSR صفر (فعال) شده باشد، اگر وقفه FIQ توسط سخت افزار جنبی تریگر شوند، مد CPU به FIQ تغییر کرده و به بردار اکسپشن FIQ در آدرس 0x0000001C پرش می کند.

در این آدرس شما باید اجرای برنامه را به زیر برنامه سرویس وقفه مربوط منشعب کنید. پس از اجرای زیر برنامه وقفه ، لازم است اطمینان حاصل کنید که پرچم وقفه مربوطه پاک شده است چرا که اگر پاک نشده باشد ، به طور پی در پی توسط سخت افزار جنبی وقفه درخواست می شود. دقت شود که پرچم وقفه با نوشتن یک بر روی آن پاک می شود و نه صفر.

وقفه برداری IRQ

در صورتی که یکی از منابع وقفه را به صورت FIQ در نظر بگیریم سایر منابع را می توانیم به ورودی های IRQ وصل کنیم. برای اطمینان از پردازش مؤثر و به موقع این وقفه ها ، VIC دارای یک جدول جستجوی (Lookup Table) قابل برنامه ریزی است که آدرس روتین سرویس وقفه یا ISR (Interrupt Service Routine) تمام وقفه در آن ذخیره می شود. VIC دارای 16 اسلات (Slot) برای آدرس دهی برداری وقفه ها است که هر اسلات شامل یک رجیستر کنترل بردار وقفه و یک رجیستر آدرس بردار است.

رجیستر کنترل بردار وقفه یا VICVectCntl (VIC Vector Control) شامل دو Field هستند ؛ فیلد کانال (بیت های 0 تا 4) و بیت فعال ساز (بیت 5). با مقدار دهی فیلد کانال هر منبع وقفه به یکی از Slot های VIC وصل شده و سپس با بیت فعال ساز ، فعال می شود. اولویت وقفه ها با شماره اسلات متناظر با آن ها مشخص می شود ، شماره کمتر به معنای اولویت بیشتر است. به عنوان مثال چنانچه بخواهیم در قطعه LPC 2138 اسلات 0 را به وقفه TIMER0 اختصاص دهیم و سپس آن را فعال کنیم ؛ دستر زیر را اجرا می کنیم:



هر اسلات VIC ، رجیستر دیگری به نام آدرس بردار یا VICVectAddr (VIC Vector Address) دارد. همانطور که از نام آن مشخص است این رجیستر باید با آدرس ISR وقفه ای که به آن اسلات اختصاص داده شده مقداردهی شود. در عمل ، زمانی که یک وقفه برداری تریگر می شود ، آدرس ISR آن که در یکی از رجیستر های VICVectAddr0 تا VICVectAddr15 قرار دارد ، در رجیستری به نام VICVectAddr کپی می شود. بنابراین زمانی که یک وقفه به صورت برداری تعریف می شود ، با درخواست آن وقفه ، آدرس ISR مربوط به آن در بخش مشخصی از حافظه به نام VICVectAddr در دسترس است.

پس از اجرای ISR وقفه برداری IRQ ، همانند وقفه FIQ لازم است پرچم وضعیت آن وقفه را پاک کنید. علاوه بر این قبل از خروج از ISR ، باید یک مقدار بی اهمیت (مانند 0x00000000) در رجیستر VICVectAddr نوشته شود تا پایان ISR وقفه جاری به VIC اعلام شده و وقفه های دیگری که در صف هستند رسیدگی می شود.

وقفه IRQ غیر برداری:

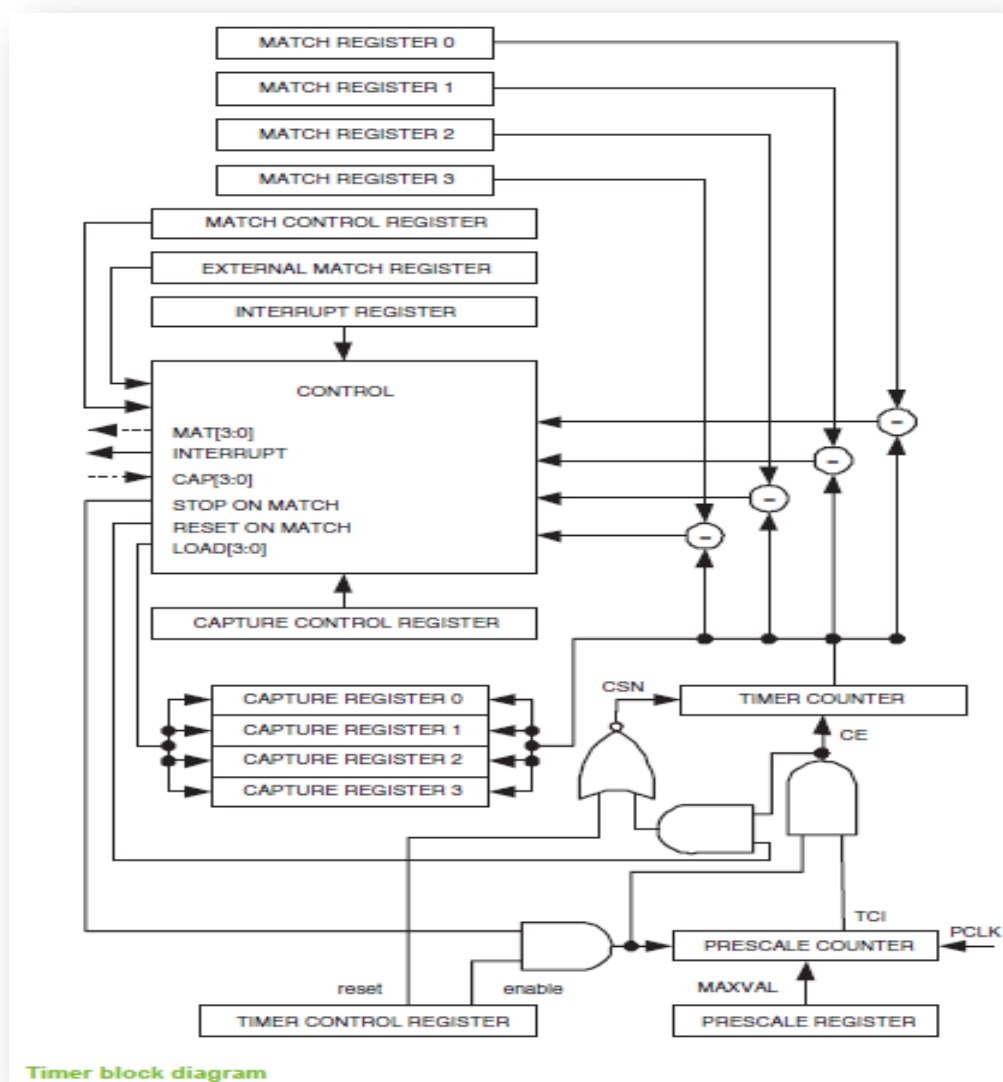
واحد VIC قادر است 16 وقفه را به صورت IRQ برداری و یکی را به صورت FIQ مدیریت کند. در صورتی که بیش از 17 منبع وقفه در میکروکنترلر فعال باشد ، VIC می تواند به صورت غیر برداری به آن ها رسیدگی کند. تمام منابع غیر برداری دارای یک ISR مشترک هستند که آدرس آن ها باید در رجیستر VICDefVectAddr (VIC Default Vector Address) ذخیره شود.

در صورتی که وقفه ای در رجیستر VICIntEnable فعال شده باشد و به عنوان FIQ تنظیم نشده باشد (بیت مربوط به آن در VICIntSelect ست نشده باشد) و اسلاتی از VIC به آن اختصاص داده نشده باشد، به عنوان وقفه IRQ غیر برداری محسوب می شود. زمانی که این وقفه درخواست می شود ، آدرس ذخیره شده در VICDefVectAddr در VICVectAddr بارگذاری شده و موجب می شود تا CPU به این آدرس پرش کند. در آنجا برنامه با خواندن رجیستر وضعیت IRQ یا VICIRQStatus (VIC IRQ Status) وقفه ای را که باعث بروز اکسپشن شده شناسایی کرده و زیر برنامه مربوط به آن را اجرا می کند.

در واقع تفاوت عمده وقفه IRQ برداری و غیر برداری در همین مسئله است که در برداری نیاز به شناسایی منبع وقفه به صورت نرم افزاری نیست و میکروکنترلر به صورت سخت افزاری به روتین وقفه درخواست شده منسحب می شود در حالی که در روش غیر برداری لازم است این موارد به صورت نرم افزاری انجام شود که طبیعتاً بسیار کندتر از روش نخست است.

تایمر/کانترهای با کاربرد عمومی:

تمام میکروکنترلرهای خانواده LPC2000 دارای چند تایمر / کانتر با کاربرد عمومی (General Purpose) هستند. تعداد دقیق این تایمر/کانترها بستگی به قطعه مورد نظر دارد اما تمام آن ها حداقل 2 تایمر / کانتر 32 بیتی دارند. به عنوان مثال قطعه LPC2138 دارای 2 تایمر / کانتر به نام های TIMER0 و TIMER1 است که این 2 کاملاً مشابه اند و تنها آدرس رجیسترهای آن ها با یکدیگر تفاوت دارد.



Timer block diagram

کلاک این تایمر/کانترها می تواند از یکی از 2 منبع PCLK و یا یکی از پایه های CAPx.n تامین شود. در حالت نخست ، تایمر/کانتر در مد تایمر کار می کند و در حالت دوم در نقش یک کانتر خواهد بود که رویدادهای خارج از تراشه را شمارش می کند.

تایمر بر مبنای یک شمارنده 32بیتی که در ورودی کلاک آن یک پیش تقسیم کننده 32 بیتی قرار گرفته است ، عمل می کند. نرخ کلاک تایمر به وسیله مقدار ذخیره شده در رجیستر پیش تقسیم کننده یا TxPR (Prescaler Register) تعیین می شود که X بسته به تایمر مورد نظر یکی از اعداد 0 یا 1 است. رجیستر شمارنده پیش تقسیم کننده یا TxPC (Prescaler Counter) با هر سیکل کلاک PCLK یک واحد افزایش می یابد تا زمانی که مقدار آن به TxPR برسد. با رسیدن TxCR به TxPR ، رجیستر شمارنده تایمر یا TxTC (Timer Counter) یک واحد افزایش یافته و رجیستر TxPC به صفر ریست می شود. تایمر به وسیله رجیستر کنترل تایمر TxTCR (Timer Control Register) کنترل می شود.

TxTCR دارای 2 بیت است که از طریق آن ها می توان تایمر را فعال ، غیر فعال و یا ریست کرد.



در حالت کانتر ، پالس ورودی از طریق یکی از پایه های CAP0.n (تایمر/کانتر 0) یا CAP1.n (تایمر/کانتر 1) دریافت می شود که n عددی بین 0 تا 3 است. در واقع ، هر تایمر/کانتر دارای 4 کانال ورودی جهت شمارش رویدادهای خارج از تراشه است. البته این 4 پین می تواند در زمان استفاده از قابلیت شکار (Capture) نیز استفاده شوند. اینکه تایمر/کانتر در کدامیک از مدهای تایمر یا کانتر کار کند بستگی به بیت های 1:0 از رجیستر کنترل شمارش یا TxCTCR (Count Control Register) دارد. در صورتی که تایمر/کانتر در وضعیت کانتر تنظیم شود به وسیله بیت های 2:3 رجیستر TxCTCR می توان یکی از 4 کانال ورودی را انتخاب کرد. علاوه بر این ، هر تایمر/کانتر دارای 4 کانال تطبیق است. هر کانال دارای یک رجیستر تطبیق به نام TxMRn (Match Register) می باشد (x بسته به تایمر/کانتر ، عدد 0 یا 1 و n عددی بین 0 تا 3 است.) که عددی 32 بیتی در آن ذخیره می شود. مقدار فعلی رجیستر شمارنده تایمر (TxTC) همواره با مقدار رجیسترهای تطبیق (TxMRn) مقایسه می شود. در صورتی که این 2 برابر باشند ، سخت افزار می تواند تایمر/کانتر را ریست یا متوقف کرده و یا وقفه ایجاد کند ؛ همچنین می تواند سطح منطقی یکی از پایه های خروجی تایمر/کانتر را ست ، ریست یا معکوس (Toggle) کند.

- Up to four external outputs corresponding to match registers, with the following capabilities:
 - Set low on match.
 - Set high on match.
 - Toggle on match.
 - Do nothing on match.

پایه های خروجی تایمر کانتر ، MATx.n هستند که x بسته به تایمر/کانتر ، عدد 0 یا 1 و n عددی بین 0 تا 3 است.

با تنظیم رجیستر کنترل تطبیق یا TxMCR (Match Control Register) می توان تعیین کرد که در زمان برابر شدن رجیستر تایمر/کانتر و رجیستر تطبیق ، چه عملی انجام شود. برای آگاهی از جزئیات رجیسترهای تایمر/کانتر به User Manual LPC2138 مراجعه شود.

واحد UART :

میکروکنترلر سری LPC2000 حداقل 2 و حداکثر 4 واحد "فرستنده گیرنده عمومی غیر همگام" یا UART (Transmitter/Universal Asynchronous Receiver) دارند. به عنوان مثال قطعات LPC2138 دارای دو UART به نام های UART0 و UART1 هستند. که از نظر کاربرد هر دو مشابه اند با این تفاوت که UART1 از مودم پشتیبانی می کند. این UART ها بر مبنای استاندارد "550" طراحی شده اند. استاندارد "550" به معماری و ساختار تراشه 16550 ساخت شرکت نشنال سیمی کاندکتور (National semiconductor) اطلاق می شود



که به طور وسیعی در پورت سریال رایانه شخصی یا PC استفاده می شود. هر UART 2 مطابق با این استاندارد دارای واحد تولید نرخ کلاک یا BoudRate و 16 Byte FIFO ارسال و دریافت هستند. از آنجایی که UART یک فرستنده/گیرنده عمومی است از آن می توان برای ارتباط با تراشه های دیگر، PC و سیستم های کنترلی (مانند PLC یا درایورهای صنعتی) به صورت تک پایانه ای (مانند RS232) و تفاضلی (مانند RS485) استفاده کرد. UART های میکروکنترلرهای LPC2000 قابلیت های فراوانی دارند.

UART دارای واحد تولید Baud Rate هستند. مولد Baud Rate یک پیش تقسیم کننده 16 بیتی است که باید کلاک PCLK را به عددی تقسیم کند که نتیجه تقسیم 16 برابر Baud Rate مورد نظر باشد (علت این است که سخت افزار UART از هر بیت دریافتی 16 بار نمونه برداری می کند):

$$UART_{baudrate} = \frac{PCLK}{16 \times (256 \times U0DLM + U0DLL)}$$

مقدار مقسم در 2 رجیستر به نام های UXDLL (UARTx Divisor Latch LSB) و UXDLM (UARTx Divisor) مقدار مقسم در 2 رجیستر به نام های UXDLL (UARTx Divisor Latch MSB) نگه داری می شود. (x بسته به UART یکی از 2 عدد 0 یا 1 است). هشت بیت نخست هر یک از این 2 رجیستر بخشی از عدد مقسم را نگه داری می کند، به عنوان مثال اگر $PCLK = 1 \text{ MHz}$ و Baud Rate = 1200 باشد:

$$1200 = \frac{1 \text{ MHz}}{16 \times (256 \times U0DLM + U0DLL)}$$

$$(256 \times U0DLM + U0DLL) = \frac{1 \text{ MHz}}{19200} \approx 52 = 0x0034$$

بنابراین مقدار 0x00 در UXDLM و 0x34 در UXDLL بارگذاری می شود.

رجیسترهای UXDLM و UXDLL دارای یک بیت محافظ به نام DLAB (Divisor Latch Access Bit) در رجیستر UXLCR (UARTx Line Control Register) است که بیش از تغییر مقدار UXDLM و UXDLL لازم است این بیت SET شده و پس از بارگذاری مقدار مورد نظر، پاک شود.

همانطور که در شکل Package LPC2138 نشان داده شده هر پایه میکروکنترلر دارای چندین عملکرد است که با توجه به رجیستر PINSEL می توان عملکرد آن را مشخص نمود اما به صورت پیش فرض عملکرد پایه ها در وضعیت GPIO (General Purpose Input/Output) قرار دارد شکل های زیر عملکردهای مختلف پایه های میکروکنترلر را نشان می دهد:



: PINSEL0

Pin function Select register 0 (PINSEL0 - address 0xE002 C000) bit description				
Bit	Symbol	Value	Function	Reset value
1:0	P0.0	00	GPIO Port 0.0	0
		01	TXD (UART0)	
		10	PWM1	
		11	Reserved	
3:2	P0.1	00	GPIO Port 0.1	0
		01	RxD (UART0)	
		10	PWM3	
		11	EINT0	
5:4	P0.2	00	GPIO Port 0.2	0
		01	SCL0 (I ² C0)	
		10	Capture 0.0 (Timer 0)	
		11	Reserved	
7:6	P0.3	00	GPIO Port 0.3	0
		01	SDA0 (I ² C0)	
		10	Match 0.0 (Timer 0)	
		11	EINT1	
9:8	P0.4	00	GPIO Port 0.4	0
		01	SCK0 (SPI0)	
		10	Capture 0.1 (Timer 0)	
		11	AD0.6	
11:10	P0.5	00	GPIO Port 0.5	0
		01	MISO0 (SPI0)	
		10	Match 0.1 (Timer 0)	
		11	AD0.7	
13:12	P0.6	00	GPIO Port 0.6	0
		01	MOSI0 (SPI0)	
		10	Capture 0.2 (Timer 0)	
		11	Reserved ^{[1][2]} or AD1.0 ^[3]	
15:14	P0.7	00	GPIO Port 0.7	0
		01	SSEL0 (SPI0)	
		10	PWM2	
		11	EINT2	
17:16	P0.8	00	GPIO Port 0.8	0
		01	TXD UART1	
		10	PWM4	

Pin function Select register 0 (PINSEL0 - address 0xE002 C000) bit description				
Bit	Symbol	Value	Function	Reset value
19:18	P0.9	00	GPIO Port 0.9	0
		01	RxD (UART1)	
		10	PWM6	
		11	EINT3	
21:20	P0.10	00	GPIO Port 0.10	0
		01	Reserved ^{[1][2]} or RTS (UART1) ^[3]	
		10	Capture 1.0 (Timer 1)	
		11	Reserved ^{[1][2]} or AD1.2 ^[3]	
23:22	P0.11	00	GPIO Port 0.11	0
		01	Reserved ^{[1][2]} or CTS (UART1) ^[3]	
		10	Capture 1.1 (Timer 1)	
		11	SCL1 (I ² C1)	
25:24	P0.12	00	GPIO Port 0.12	0
		01	Reserved ^{[1][2]} or DSR (UART1) ^[3]	
		10	Match 1.0 (Timer 1)	
		11	Reserved ^{[1][2]} or AD1.3 ^[3]	
27:26	P0.13	00	GPIO Port 0.13	0
		01	Reserved ^{[1][2]} or DTR (UART1) ^[3]	
		10	Match 1.1 (Timer 1)	
		11	Reserved ^{[1][2]} or AD1.4 ^[3]	
29:28	P0.14	00	GPIO Port 0.14	0
		01	Reserved ^{[1][2]} or DCD (UART1) ^[3]	
		10	EINT1	
		11	SDA1 (I ² C1)	
31:30	P0.15	00	GPIO Port 0.15	0
		01	Reserved ^{[1][2]} or RI (UART1) ^[3]	
		10	EINT2	
		11	Reserved ^{[1][2]} or AD1.5 ^[3]	

[1] Available on LPC2131 and LPC2131/01.
 [2] Available on LPC2132 and LPC2132/01.
 [3] Available on LPC2134/6/8, LPC2134/01, LPC2136/01, and LPC2138/01.



: PINSEL1

Pin function Select register 1 (PINSEL1 - address 0xE002 C004) bit description				
Bit	Symbol	Value	Function	Reset value
1:0	P0.16	00	GPIO Port 0.16	0
		01	EINT0	
		10	Match 0.2 (Timer 0)	
		11	Capture 0.2 (Timer 0)	
3:2	P0.17	00	GPIO Port 0.17	0
		01	Capture 1.2 (Timer 1)	
		10	SCK (SSP)	
		11	Match 1.2 (Timer 1)	
5:4	P0.18	00	GPIO Port 0.18	0
		01	Capture 1.3 (Timer 1)	
		10	MISO (SSP)	
		11	Match 1.3 (Timer 1)	
7:6	P0.19	00	GPIO Port 0.19	0
		01	Match 1.2 (Timer 1)	
		10	MOSI (SSP)	
		11	Capture 1.2 (Timer 1)	
9:8	P0.20	00	GPIO Port 0.20	0
		01	Match 1.3 (Timer 1)	
		10	SSEL (SSP)	
		11	EINT3	
11:10	P0.21	00	GPIO Port 0.21	0
		01	PWMS	
		10	Reserved ^[1] or AD1.6 ^[3]	
		11	Capture 1.3 (Timer 1)	
13:12	P0.22	00	GPIO Port 0.22	0
		01	Reserved ^[1] or AD1.7 ^[3]	
		10	Capture 0.0 (Timer 0)	
		11	Match 0.0 (Timer 0)	
15:14	P0.23	00	GPIO Port 0.23	0
		01	Reserved	
		10	Reserved	
		11	Reserved	
17:16	P0.24	00	Reserved	0
		01	Reserved	
		10	Reserved	
		11	Reserved	
19:18	P0.25	00	GPIO Port 0.25	0
		01	AD0.4	
		10	Reserved ^[1] or Aout(DAC) ^{[2][3]}	
		11	Reserved	

Pin function Select register 1 (PINSEL1 - address 0xE002 C004) bit description				
Bit	Symbol	Value	Function	Reset value
21:20	P0.26	00	GPIO Port 0.26	0
		01	AD0.5	
		10	Reserved	
		11	Reserved	
23:22	P0.27	00	GPIO Port 0.27	0
		01	AD0.0	
		10	Capture 0.1 (Timer 0)	
		11	Match 0.1 (Timer 0)	
25:24	P0.28	00	GPIO Port 0.28	0
		01	AD0.1	
		10	Capture 0.2 (Timer 0)	
		11	Match 0.2 (Timer 0)	
27:26	P0.29	00	GPIO Port 0.29	0
		01	AD0.2	
		10	Capture 0.3 (Timer 0)	
		11	Match 0.3 (Timer 0)	
29:28	P0.30	00	GPIO Port 0.30	0
		01	AD0.3	
		10	EINT3	
		11	Capture 0.0 (Timer 0)	
31:30	P0.31	00	GPO Port only	0
		01	Reserved	
		10	Reserved	
		11	Reserved	

[1] Available on LPC2131 and LPC2131/01.

[2] Available on LPC2132 and LPC2132/01.

[3] Available on LPC2134/6/8, LPC2134/01, LPC2136/01, and LPC2138/01.



: PINSEL2

Pin function Select register 2 (PINSEL2 - 0xE002 C014) bit description				
Bit	Symbol	Value	Function	Reset value
1:0	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	GPIO/DEBUG	0	Pins P1.36-26 are used as GPIO pins.	P1.26/RTCK
		1	Pins P1.36-26 are used as a Debug port.	
3	GPIO/TRACE	0	Pins P1.25-16 are used as GPIO pins.	P1.20/ TRACESYNC
		1	Pins P1.25-16 are used as a Trace port.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

و برای استفاده از UART0 لازم است رجیستر PINSEL0 با عدد 0x00000003 مقداردهی شود تا پایه های P0.0 و P0.2 به ترتیب به TxD0 و RxD0 متصل شوند. و برای استفاده از UART1 لازم است رجیستر PINSEL0 با عدد 0x00050000 مقداردهی شود تا پایه های P0.8 و P0.9 به ترتیب به TxD1 و RxD1 متصل شوند. در ارتباطات بر مبنای UART برخی مشخصه های قاب داده (Data Frame) می تواند توسط کاربر تعیین شود.

این مشخصه ها عبارتند از تعداد بیت های داده (5، 6، 7 و یا 8)، تعداد بیت های پایان (1 یا 2)، بیت توازن (فرد، زوج یا غیر فعال)، و ارسال سیگنال توقف (Break، سیگنال توقف در استاندارد RS-232 تعریف شده است و با ارسال چند بیت صفر متوالی (حالت بیکاری خط، یک منطقی است.) ایجاد می شود.) این تنظیمات توسط بیت های رجیستر UxLCR انجام می شود. به عنوان مثال با انتساب مقدار 0x03 به رجیستر U1LCR در وضعیت "داده 8 بیتی"، "1 بیت پایان"، "بدون توازن و سیگنال توقف پیکربندی می شود."

UART0 Line Control Register (U0LCR - address 0xE000 C00C) bit description				
Bit	Symbol	Value	Description	Reset value
1:0	Word Length Select	00	5 bit character length	0
		01	6 bit character length	
		10	7 bit character length	
		11	8 bit character length	
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if U0LCR[1:0]=00).	
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		10	Forced "1" stick parity.	
		11	Forced "0" stick parity.	
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UART0 TXD is forced to logic 0 when U0LCR[6] is active high.	
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	



UART1 Line Control Register (U1LCR - address 0xE001 000C) bit description

Bit	Symbol	Value	Description	Reset value
1:0	Word Length Select	00	5 bit character length.	0
		01	6 bit character length.	
		10	7 bit character length.	
		11	8 bit character length.	
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if U1LCR[1:0]=00).	
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		10	Forced "1" stick parity.	
		11	Forced "0" stick parity.	
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UART1 TXD is forced to logic 0 when U1LCR[6] is active high.	
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	

پس از انجام تنظیمات اولیه UART ، می توانیم با نوشتن داده بر روی رجیستر UxTHR (UARTx Transmitter Holding Register) یک کاراکتر را ارسال نمود. به طور مشابه با خواندن رجیستر UxRBR (UARTx Receive Buffer Register) کاراکتر موجود در بافر UART دریافت می شود. در حقیقت این دو رجیستر هر دو یک محل از حافظه را اشغال می کنند و آدرس مشترک 0xE0010000 دارند ، زمانی که یک کاراکتر در این محل نوشته می شود به طور خودکار سخت افزار آن را به انتهای FIFO ارسال می کند و خواندن از این آدرس ، داده نخست از FIFO دریافت را به دست می دهد.

بخش فرستنده UART دارای یک پرچم به نام THRE (Transmitter Holding Register Empty) در رجیستر UxLSR (Line Status Register) است که یک بودن آن نشان دهنده این است که بافر ارسال خالی است و امکان نوشتن داده در رجیستر UxTHR وجود دارد. در نقطه مقابل، بخش گیرنده نیز پرچم مشابهی به نام RDR (Receiver Data Register) در رجیستر UxLSR دارد که با یک شدن آن می توانیم داده دریافت شده را بخوانیم.



برنامه Source در نرم کامپایلر Keil :

```
#include <lpc213x.h>
#include <stdio.h>
#include "lcd.h"

int h=0,m=0,s=0,y=0,t[3],w=0,set,up,down;
char lcd[80],b[16],c[16];
unsigned int x=0,test=0;
unsigned long int z=0;
char person=0;

__irq void TMR0(void)
{
    s++;
    y=0;
    w=0;
    if(b[0]=='2'&&b[1]=='5'&&b[2]=='0'&&b[3]=='0'&&b[4]=='D'
        &&b[5]=='1'&&b[6]=='5'&&b[7]=='1'&&b[8]=='5'&&b[9]=='7'
        &&b[10]=='F'&&b[11]=='2') person=1;
    if(b[0]=='2'&&b[1]=='5'&&b[2]=='0'&&b[3]=='0'&&b[4]=='D'
        &&b[5]=='1'&&b[6]=='6'&&b[7]=='2'&&b[8]=='6'&&b[9]=='8'
        &&b[10]=='F'&&b[11]=='E') person=2;
    if(b[0]=='2'&&b[1]=='5'&&b[2]=='0'&&b[3]=='0'&&b[4]=='D'
        &&b[5]=='1'&&b[6]=='5'&&b[7]=='9'&&b[8]=='9'&&b[9]=='1'
        &&b[10]=='3'&&b[11]=='C') person=3;
    TOIR=0x1;
    TOTC=0xFFB9B000;
    VICVectAddr=0;
}

__irq void U0(void)
{
    t[0]=U0IIR;
    t[1]=U0LSR;
    if(U0LSR & 0x01)
    {
        c[w]=U0RBR;
        w++;
        c[w]=0;
    }
    U0TER=0;
    U0IER=0x5;
    U0FCR=0x07;
    VICVectAddr=0;
}
```



```
__irq void U1(void)
{
    t[0]=U1IIR;
    t[1]=U1LSR;
    if(U1LSR & 0x01)
    {
        b[y]=U1RBR;
        y++;
        b[y]=0;
    }
    U1TER=0;
    U1IER=0x5;
    U1FCR=0x07;
    VICVectAddr=0;
}

unsigned char getkey (void)
{
    while (!(U0LSR & 0x01));
    return (U0RBR);
}

void sendchar(char ch)
{
    U0THR=ch;
    U0TER=0x80;
    while (!(U0LSR & 0x20));
    for(z=0; z<20000; z++);
}

void send(char *p,char enter)
{
    if(enter==2) sendchar(34);
    while(*p!=0){sendchar(*p); *p++;}
    if(enter==1) sendchar(13);
    if(enter==2) {sendchar(34); sendchar(13);}
    for(z=0; z<2000000; z++);
    if(test==0)
    {
        test=1;
        IO0DIR=0X40;
        IO0SET=0X40;
    }
    else
    {
        test=0;
        IO0DIR=0X40;
        IO0CLR=0X40;
    }
}
```



```
void setting()
{
    VICIntEnClr=0x000000D0;
    do
    {
        set=IOOPIN;
        set=set&0x08;
    }while(set==0);
    lcd_clear();
    displayText("Hour Setting");
    for(z=0;z<200000;z++);
    do
    {
        set=IOOPIN;
        set=set&0x08;
        up=IOOPIN;
        up=up&0x10;
        down=IOOPIN;
        down=down&0x20;
        if(up==0 && h<59) h++;
        if(down==0 && h>0) h--;
        lcd_clear();
        sprintf(lcd,"%02u ",h);
        displayText(lcd);
        for(z=0;z<100000;z++);
    }while(set==0x08);
    do
    {
        set=IOOPIN;
        set=set&0x08;
    }while(set==0);
    lcd_clear();
    displayText("minute Setting");
    for(z=0;z<200000;z++);
    do
    {
        set=IOOPIN;
        set=set&0x08;
        up=IOOPIN;
        up=up&0x10;
        down=IOOPIN;
        down=down&0x20;
        if(up==0 && m<59) m++;
        if(down==0 && m>0) m--;
        lcd_clear();
        sprintf(lcd,"%02u ",m);
        displayText(lcd);
        for(z=0;z<500000;z++);
    }while(set==0x08);
    s=0;
    VICIntEnable=0x000000D0;
}
}
```



```
int main(void)
{
    VICVectCntl0=(0x20|4); //TIMER0
    VICVectCntl1=(0x20|6); //UART0
    VICVectCntl2=(0x20|7); //UART1
    VICVectAddr=0;
    VICVectAddr0=(unsigned long int) &TMRO;
    VICVectAddr1=(unsigned long int) &U0;
    VICVectAddr2=(unsigned long int) &U1;
    PINSEL0=0x00050005;
    PINSEL1=0;
    PINSEL2=0;
    lcd_init();
    TOTCR=0x1;
    TOMR0=0xFFFFFFFF;
    TOIR=0x1;
    TOMCR=0x1;
    TOTC=0xFFB9B000;
    //U0FDR=0xB1;
    U0LCR=0x83;
    U0DLL=30;
    U0DLM=0;
    U0LCR=0x03;
    U0IER=0x5;
    U0FCR=0x07;

    U1LCR=0x83;
    U1DLL=30;
    U1DLM=0;
    U1LCR=0x03;
    U1IER=0x5;
    U1FCR=0x07;
    VICIntEnable=0x000000D0;
    test=0;
    while(1)
    {
        if(s>59)
        {
            s=0;
            m++;
        }
        if(m>59)
        {
            m=0;
            h++;
        }
        if(h>23) h=m=s=0;
        lcd_gotoxy(0,0);
        sprintf(lcd,"%02u:%02u:%02u ",h,m,s);
        displayText(lcd);
        for(z=0;z<500000;z++);
        lcd_gotoxy(10,0);
        displayText(c);
        //lcd_clear();
    }
}
```



```
if(test==0)
{
    test=1;
    IOODIR=0X40;
    IOOSET=0X40;
}
else
{
    test=0;
    IOODIR=0X40;
    IOOCLR=0X40;
}
//IOODIR=0;
set=IOOPIN;
set=set&0x08;
if(set==0) setting();
//person=1;
if(person>0)
{
    //VICIntEnClr=0x000000D0;
    lcd_gotoxy(0,0);
    displayText("Sending...");
    lcd_gotoxy(0,1);
    if(person==1) displayText("Tahoonchi");
    if(person==2) displayText("Hosseini");
    if(person==3) displayText("Beygi");
    send("AT",1);
    send("AT",1);
    send("AT",1);
    send("AT+CMGF=1",1);
    send("AT+CMGD=1",1);
    send("AT+CMGS=",0);
    if(person==1) send("09375269805",2);
    if(person==2) send("09363140825",2);
    if(person==3) send("09124193070",2);
    w=0;
    //send("welcome",0);
    if(person==1)
    {
        sprintf(lcd,"%02u:%02u:%02u Tahoonchi Is Present",h,m,s);
        send(lcd,3);
    }
    if(person==2)
    {
        sprintf(lcd,"%02u:%02u:%02u Hosseini Is Present",h,m,s);
        send(lcd,3);
    }
    if(person==3)
    {
        sprintf(lcd,"%02u:%02u:%02u Beygi Is Present",h,m,s);
        send(lcd,3);
    }
}
```

Melec.ir



```
        /*do
        {
            x=IOOPIN;
            x=x&0x20;
        }while(x==0);*/
        sendchar(26);
        //sendchar(26);
        //VICIntEnable=0x000000D0;
        for(person=0;person<16;person++) b[person]=0;
        VICVectAddr=0;
        person=0;
        lcd_clear();
    }
}
}
```

برنامه LCD در کامپایلر Keil :

```
#define LCD_DATA_DIR    IO0DIR
#define LCD_DATA_SET    IO0SET
#define LCD_DATA_CLR    IO0CLR
#define LCD_DATA_PIN    IO0PIN
#define LCD_DATA_MASK    0x7E600000
#define BUSY            (1<<21)
#define RS (1<<30)
#define RW (1<<29)
#define EN (1<<28)
#define D0 IO1SET=(1<<17)
#define D1 IO0SET=(1<<27)
#define D2 IO0SET=(1<<26)
#define D3 IO0SET=(1<<25)
#define D4 IO1SET=(1<<18)
#define D5 IO1SET=(1<<19)
#define D6 IO0SET=(1<<22)
#define D7 IO0SET=(1<<21)

void lcd_put_data(unsigned char data)
{
    unsigned char temp[8],i=0;
    IO0CLR=0x0E600000;
    IO1CLR=0xFFFFFFFF;
    i=0;
    while(i<8)
    {
        temp[i]=data%2;
        data=data/2;
        i++;
    }
    if(temp[0]==1) D0;
    if(temp[1]==1) D1;
    if(temp[2]==1) D2;
    if(temp[3]==1) D3;
```



```
    if(temp[4]==1) D4;
    if(temp[5]==1) D5;
    if(temp[6]==1) D6;
    if(temp[7]==1) D7;
    LCD_DATA_SET=EN;
    for(i=0; i<10; i++);
    LCD_DATA_CLR=EN;
    IO0CLR=0x0E600000;
    IO1CLR=0x000E0000;
}

void checkBusy()
{
    unsigned long int x;
    IO0DIR=0x70000000;
    IO1DIR=0;
    LCD_DATA_CLR=RS;
    LCD_DATA_SET=RW;
    do
    {
        LCD_DATA_SET=EN;
        for(x=0; x<10; x++);
        x=IO0PIN;
        x=x&0x00200000;
        LCD_DATA_CLR=EN;
    }while(x==0x00200000);
    IO0DIR=0x7E600000;
    IO1DIR=0x000E0000;
}

void writeCommand(unsigned char data)
{
    checkBusy();
    LCD_DATA_CLR=RS;
    LCD_DATA_CLR=RW;
    lcd_put_data(data);
}

void writeData(unsigned char data)
{
    checkBusy();
    LCD_DATA_SET=RS;
    LCD_DATA_CLR=RW;
    lcd_put_data(data);
}

void lcd_init(void)
{
    IO0DIR=0x7E600000;
    IO1DIR=0x000E0000;
    writeCommand(0x01);
    writeCommand(0x38);
    writeCommand(0x0C);
}
```

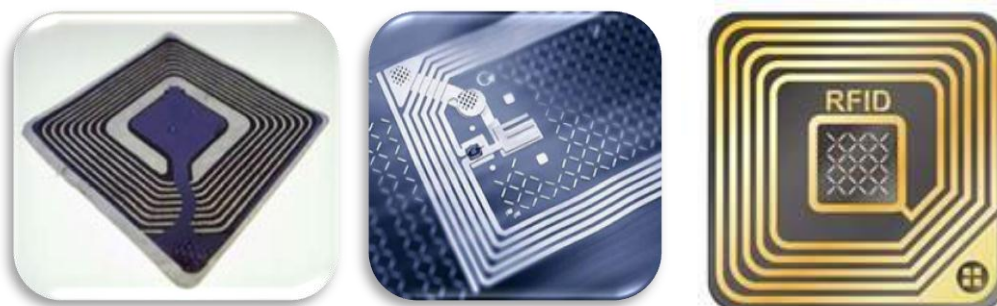


```
void displayText(char *p)
{
  while(*p!='\0')writeData(*(p++));
}

void lcd_gotoxy(unsigned char x,unsigned char y)
{
  if ((x<40)&&(y<2))
  {
    if (y==0) writeCommand(0x80 + x);
    else writeCommand(0xC0 + x);
  }
}

void lcd_clear(void)
{
  checkBusy();
  LCD_DATA_CLR=RS;
  LCD_DATA_CLR=RW;
  writeCommand(0x01);
}
```

مقدمه ای بر تاریخچه RFID



RFID (Radio Frequency Identification) به معنی ابزار تشخیص امواج رادیویی است. RFID دستگاه الکترونیکی کوچکی است که شامل یک تراشه کوچک و یک آنتن می باشد. این تراشه قادر به حمل 2000 بایت اطلاعات یا کمتر می باشد. برای روشنتر شدن مطلب می توان گفت دستگاه RFID کاربردی شبیه بارکد و نوارهای مغناطیسی نصب شده روی credit card ها یا کارتهای ATM دارد. RFID برای هر شیء یک مشخصه واحد ایجاد می کند که از دیگر اشیا قابل شناسایی خواهد شد و همین طور که از روی بارکد یا نوار مغناطیسی می توان اطلاعات را خواند RFID هم میتواند خوانده شده واز آن طریق اطلاعات آن دریافت یا اصلاح شود. در واقع RFID یک عبارت کلی است برای تشریح سیستمی که هویت (در قالب یک شماره سریال منحصر به فرد) یک شیء یا انسان را از راه دور با استفاده از امواج رادیویی ارسال می کند. این سیستم بخشی از تکنولوژی های خودکار شناسایی است.



تکنولوژی‌های خودکار شناسایی یا Auto-ID Technologies شامل بارکدها، دستگاه رمزخوان اپتیکال و برخی از تکنولوژی‌های زیست‌سنجی (Biometrics) از جمله اسکنرهای قرنیه چشم می‌شود. از این تکنولوژی‌ها در جهت کاهش زمان و نیروی انسانی لازم برای وارد کردن اطلاعات به صورت دستی و افزایش دقت اطلاعات استفاده می‌شود.

• TAG

معمولترین تگ‌ها برای استفاده در انبارها و مراکز توزیع، برچسب‌های غیرفعال یا برچسب‌های هوشمند (Smart Labels) است که به بسته‌ها و یا پالت‌ها الصاق می‌شوند. یک برچسب هوشمند نمونه دارای یک تگ RFID است (که در داخل آن قرارداد شده) و کد بارکد و متن مناسب نیز برای پشتیبانی سیستم‌های قدیمی مدیریت انبار بر روی آن چاپ شده است. آنتن و تراشه RFID همچنین می‌تواند در پوشش‌های مقاومتر برای الصاق دائمی بر اموال، شناسایی و تعیین موقعیت و یا مقاومت در شرایط مختلف از قبیل درجه حرارت بالا، مایعات صنعتی، فشار، ضربه و شرایطی که روش‌های دیگر جمع‌آوری داده ممکن نمی‌باشد، قرار داده شود.

تگ‌خوان‌ها دارای آنتن جهت ارسال و دریافت سیگنال‌های رادیویی، یک پردازنده جهت بازخوانی اطلاعات تگ و کنترل‌کننده و نرم‌افزاری جهت انجام پردازش‌ها و عملیات پیشرفته مدیریت داده‌ها، می‌باشند. آنتن ممکن است برای نصب در محل‌های مختلف جدا از سیستم پردازنده بوده و از طریق یک کابل به آن متصل شود. انواع مختلفی از تگ‌خوان‌ها فراهم شده است؛ مانند تگ‌خوان دستی قابل حمل و یا تگ‌خوان ثابت که معمولاً در ورودی انبار و خطوط نوار نقاله نصب می‌شود.

• فرکانس‌ها و استاندارد EPC

فرکانس یکی از مهمترین پارامترهای موثر در بُرد و کارایی سیستم RFID است. تقریباً کلیه سیستم‌های RFID استفاده شده در کاربرهای انبار، مراکز توزیع و زنجیره‌های تامین در یکی از سه خانواده فرکانسی زیر می‌باشد:

• فرکانس (HF) 13.56 MHz: معمولاً برای فواصل کوتاه و در حدود یک متر.

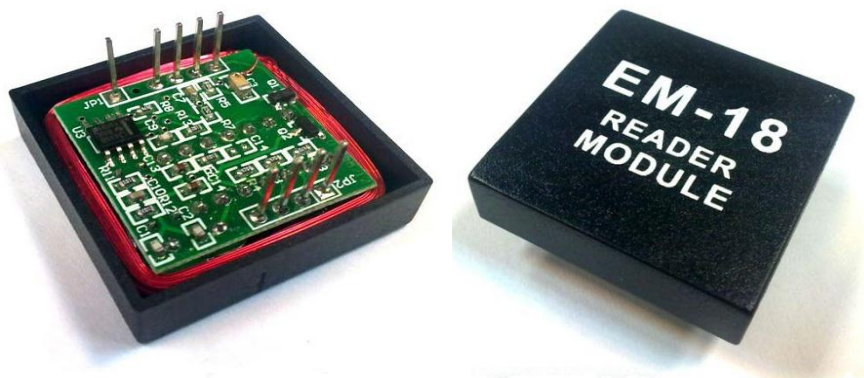
• فرکانس (UHF) 860~960 MHz: که استاندارد شناخته شده EPC-Global نسل دوم (Gen 2) را شامل می‌شود و بُرد تا 6 متر را خواهد داشت.

• فرکانس 45.2 GHz: که در تگ‌های فعال استفاده می‌شود و بُرد بالایی دارد. معمولاً در شناسایی و تعیین موقعیت در محوطه‌های بزرگ کاربرد دارد.

فناوری UHF، محدوده پوشش مناسب، سرعت، امنیت و تجهیزات مقرون‌بصرفه‌ای را که بیشتر برنامه‌های کاربردی و سیستم‌های انبار، مدیریت کالا و تولید نیاز دارند فراهم می‌نماید. استاندارد EPC Gen 2 UHF برای رفع نیازهای زنجیره‌های تامین ارائه شده است و در محدوده فرکانسی 860~960 MHz عمل می‌کند، همچنین احتمال تداخل با سیستم‌های رادیویی مانند استاندارد شبکه‌های رادیویی 802.11 کمتر می‌باشد. این استاندارد سرعت بالا در شناسایی و خواندن تگ‌ها، فاصله مناسب، کدهای الکترونیکی محصول که شماره‌گذاری سریال منحصر بفرد را فراهم می‌کند و میزان حافظه مناسبی را نیز برای ذخیره اطلاعات کاربر، ارائه می‌نماید.

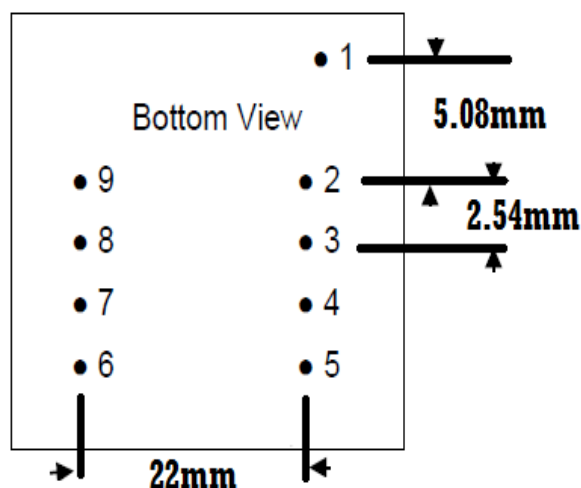


ماژول EM-18 RFID و ارتباط آن با میکروکنترلر

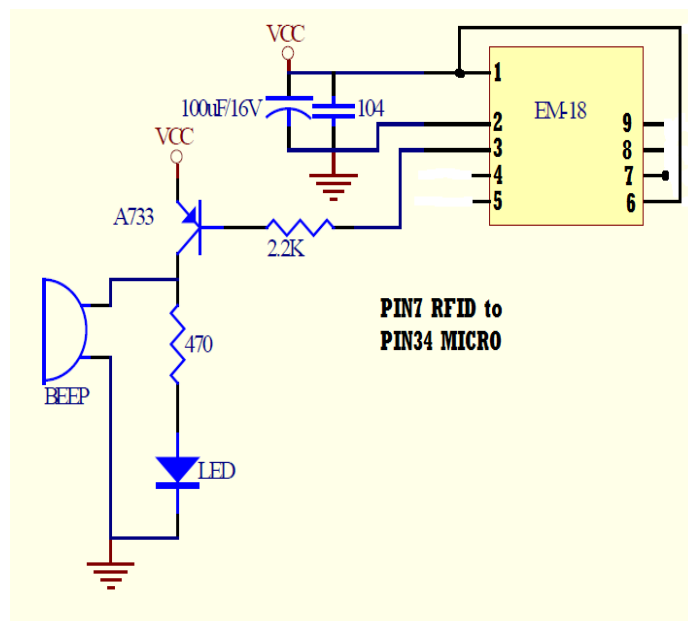


دیتا شیت پایه ها و نحوه ی بایاس پایه ها جهت کاربری مورد نظر در پروژه با توجه به جدول دیتا شیت. البته لازم به ذکر است بدلیل موجود نبودن این قطعه در کتابخانه برنامه ها ی Proteus و Altium Designer توسط خودمان به صورت کتابخانه تعریف شد.

1	VCC	5V
2	GND	GND
3	BEEP	BEEP AND LED
4	ANT	NO USE
5	ANT	NO USE
6	SEL	HIGH IS RS232, LOW IS WEIGAND
7	RS232	RS232
8	D1	WEIGAND DATA 1
9	D0	WEIGAND DATA 0



ارتباط بین ماژول EM-18 RFID با میکروکنترلر از طریق UART1 پایه ی RXD1 میکروکنترلر برقرار گردیده است و نقشه مدار آن به شکل زیر است. البته برنامه



مقدمه ای بر تاریخچه ماژول GSM (SIM 900)

GSM مخفف عبارت **Global System For Mobile Communications** یا سامانه جهانی ارتباطات همراه گسترش گوشی‌های سلولی تلفن همراه در اوایل دهه‌ی هشتاد میلادی در اروپا، رو به زوال نهاد. عدم وجود استانداردهای تکنولوژیکی، سران اجلاس وزرای ارتباطات و فناوری اطلاعات کشورهای اروپایی را در سال 1982 با هدف توسعه‌ی یک استاندارد واحد برای گوشی‌های همراه که در سراسر قاره، قابل استفاده باشد، وادار ساخت تا به ساخت گروه ویژه‌ی تلفن همراه (GSM) مبادرت ورزند.

نخستین شبکه‌ی استاندارد جی‌اس‌ام نیز در سال 1988 و در فنلاند افتتاح شد. در سال 1989، مسوولیت مدیریت استاندارد به موسسه‌ی استانداردهای ارتباطات سیار اروپا (ETSI) واگذار شد و فاز نخست راه‌اندازی شبکه‌های GSM در سراسر قاره نیز به سال 1990 بهره‌برداری شد. در پایان سال 1993، بالغ بر یک میلیون کاربر از طریق 70 سرویس دهنده در 48 کشور از سرویس‌های در سراسر اروپا از سرویس‌های جی‌اس‌ام استفاده می‌کردند .

مخابرات سلولی یکی از سریعترین Application های رو به رشد در صنعت ارتباطات است . هر روزه بر تعداد مشترکین این نوع ارتباط در جهان افزوده می شود .

تجارت ارتباطات موبایل بسرعت در CEPT (دفاتر پست و مخابرات اروپایی) در حال رشد و توسعه است . CEPT از طریق بازارهای پر قدرت موبایل ، توسعه فناوری موبایل را رهبری می نماید و همکاریهای جدیدی در زمینه سیستمهای استاندارد ساز ، پیاده سازی و اجرای این فناوریها بوجود آورده است .



یکی از مهمترین محصولات این استانداردها که در CEPT شکل گرفته است استاندارد GSM است. این استاندارد سیستم ارتباطات موبایل سلول‌ی دیجیتال‌ی نسل جدید را در CEPT اروپا توسعه داده است. برای اولین بار کار استاندارد سازی GSM جهت پیاده سازی این سیستم در سال 1991 صورت گرفته است.

CEPT=European Post offices and Telecommunication

سیستم GSM از ترکیب 3 زیر سیستم اصلی بوجود آمده است :

۱. زیر سیستم شبکه
۲. زیر سیستم رادیویی
۳. زیر سیستم پشتیبانی و نگهداری

در سیستم GSM برای برقراری ارتباطات اپراتورهای شبکه بامنابع مختلف و تجهیزات زیر ساختار سلول‌ی ، نه تنها رابطی هوایی بلکه چندین رابط اصلی دیگر برای مرتبط کردن قسمت های مختلف این سیستم تعریف شده است.

سه رابط مهم در سیستم GSM در زیر آمده است :

- رابط A که میان MSC و BSC قرار دارد .
- رابط bis-A که میان BSC و BTS قرار دارد .
- رابط UM که میان BTS و MS قرار دارد .
- رابط دیگری نیز به نام MAP وجود دارد که پروتکلی است که میان عناصر MSC ، VLR ، HLR ، EIR و AUC رد و بدل می شود .

۱. زیر سیستم شبکه

این سیستم شامل تجهیزات و فانکشنهای مربوط به مکالمات end-to-end ، مدیریت مشترکین ، Mobility می باشد و نیز مانند رابطی میان سیستم GSM و مراکز تلفن ثابت (PSTN) عمل می کند .

زیر سیستم شبکه ، یک زیر سیستم سوئیچینگ می باشد که شامل MSC ها ، VLR ، HLR ، AUC و EIR می باشد .

در زیر تعریف کوتاهی از هر یک از این عناصر ارائه شده است :

MSC : یا مرکز سرویسهای سوئیچینگ موبایل فانکشنهای راه اندازی مکالمه (call setup) را انجام می دهد ، رابطی نیز با مراکز تلفن ثابت دارد و فانکشنهایی نیز مانند ارائه صورت حساب مشترکین نیز برعهده این مرکز است .

HLR : یا ثبت کننده محل HOME یک پایگاه داده متمرکز شامل اطلاعات تمامی مشترکین ثبت شده در یک PLMN است . ممکن است در یک PLMN بیشتر از یک HLR وجود داشته باشد ولی هر مشترک مشخص تنها به یک HLR میتواند وارد شود .



VLR : یا ثبت کننده محل visitor يك پایگاه داده شامل اطلاعات موبایل هایی است که در حال حاضر در حوزه MSC ي کنترلر در حال حرکت هستند . در زمانی که يك MS به حوزه MSC جدیدی وارد می شود ، VLR ي که به آن MSC متصل شده است ، اطلاعات MS مورد نظر را از HLR درخواست می کند . HLR نیز اطلاعات MS مورد نظر را به آن MSC که MS در حوزه اش قرار دارد ، ارائه خواهد داد . اگر يك MS بخواهد مکالمه ای برقرار نماید VLR تمام اطلاعات مورد نیاز جهت برقراری مکالمه را ارائه خواهد داد و لزومی ندارد که در هر لحظه از HLR سوال نماید . VLR در يك جمله می توان گفت ، يك HLR توزیع شده است و شامل اطلاعات دقیقی در مورد محل يك موبایل است .

AUC : یا مرکز تعیین هویت به HLR متصل می شود و وظیفه آن آماده سازی HLR به همراه پارامترهای تعیین هویت و کلیدهای رمزنگاری است که این عملیات برای اهداف امنیتی استفاده می شوند .

EIR : یا ثبت کننده هویت تجهیزات يك پایگاه داده است که در آن شماره های بین المللی تعیین هویت تجهیزات موبایل (IMEI) ، برای هر دستگاه موبایل ثبت شده ، ذخیره می شود .

یکی دیگر از ترکیبات زیر سیستم شبکه Echo Canceller است که مسایل آزار دهنده ای (مانند انعکاس صدا) که از طریق شبکه موبایل در زمان اتصال به يك مدار PSTN ایجاد می شود را کاهش می دهد .

شبکه IWF یا فانکشن داخل شبکه ای نیز رابطی میان MSC و دیگر شبکه ها (PSTN و ISDN) میباشد .

۲. زیرسیستم رادیویی:

شامل تجهیزات و فانکشنهای مرتبط با مدیریت اتصالات مسیر رادیویی ، مانند مدیریت handoverها می باشد. این زیر سیستم شامل BSC ، BTS و MS است . MS بطور قراردادی در زیر سیستم رادیویی قرار گرفته و همیشه آخرین مسیر يك مکالمه است و از برقراری يك مکالمه ، به همراه زیر سیستم شبکه ، جهت مدیریت mobility ، محافظت می کند .

(IWF=InterWorking Function)

MS دارای قابلیت های پایانه شبکه و همچنین پایانه کاربر است . هر سلول در سیستم GSM يك BTS با چندین گیرنده و فرستنده دارد . يك گروه از BTS ها توسط يك BSC کنترل میشوند . پیکربندیهای مختلفی برای BSC-BTS وجود دارد . برخی از این پیکر بندیها برای وضعیت ترافیک بالا و تعدادی برای مناطقی با ترافیک متوسط طراحی شده اند . يك BSC فانکشنهایی چون handover و power control را نیز کنترل مینماید . BSC و BTS با هم بنام BSS شناخته می شوند . BSS از دید MSC بصورت يك رابط که ارتباطات لازم را با MS ها در حوزه ای مشخص برقرار می کند ، به نظر می رسد . BSS دائما با يك مدیریت کانال رادیویی ، فانکشنهای انتقال ، کنترل link رادیویی و تخمین کیفیت و مهیا سازی سیستم برای handover ها ، مرتبط است . BSS می تواند به N سلول پوشش بدهد که N می تواند يك یا بیشتر باشد .



زیرسیستم مرکز نگهداری و پشتیبانی (OMC) شامل فانکشنهای نگهداری و پشتیبانی تجهیزات GSM می باشد و پشتیبانی رابط اپراتور شبکه را نیز برعهده دارد . OMC به تمام تجهیزات داخل سیستم سوئیچینگ و BSC متصل می شود . OMC در حقیقت فانکشنهای نظارتی GSM يك کشور را انجام میدهد (مانند صورتحساب دادن) و یکی دیگر از مهمترین فانکشنهای آن هم ، فانکشن نگهداری HLR يك کشور است . بسته به سایز شبکه هر کشور میتواند بیشتر از يك OMC داشته باشد . مدیریت سراسری و متمرکز شبکه نیز توسط مرکز مدیریت شبکه (NMC) انجام میپذیرد و OMC نیز مسئول مدیریت منطقه ای شبکه میباشد . مطالب فوق شرح مختصری در مورد معماری GSM ، عناصر ، فانکشنها و رابطهای سیستم GSM بود .

: BTS

در شبکه موبایل اولین بخشی که مستقیماً با گوشی موبایل در ارتباط است به لفظ عوام آنتن موبایل و به تعبیر تخصصی Base Transceiver Station میباشد.

: TDMA

همانطور که در مباحث گذشته عنوان شد آنتن موبایل به عنوان واسطه بین گوشی موبایل با شبکه موبایل می باشد در شبکه موبایل گزارش آخرین مکان مشترک موبایل و ارسال و دریافت شماره تلفن و مسائلی از این دست به عهده کانال سیگنالینگ می باشد و بدین شکل مکالمات که هدف اصلی می باشد به عهده کانالهای ترافیکی می باشد. به طور قطع BTS وظیفه ارائه این کانالها را به مشترک موبایل دارا می باشد.

یکی از مشکلات اساسی شبکه های موبایل کمبود فضای فرکانسی می باشد که به روشهای متنوعی توانسته اند این مشکل را برطرف کنند یکی از این روشها استاندارد TDMA می باشد . در فرستندهای محلی تلویزیون و رادیو برای هر شبکه رادیویی و تلویزیونی از يك فرکانس استفاده می شود ولی در شبکه موبایل از يك فرکانس برای ارسال یا دریافت اطلاعات ۸ مشترک استفاده می شود .

:MSISDN

یکی از موارد اصلی ذخیره شده در این واحد MSISDN یا همان شماره موبایل است که در اختیار من و شما می باشد و همه با این شماره کار تماس را انجام می دهند مثل 09121620000

:IMSI

هر سیم کارت دارای يك شماره واحد و منحصر به فرد در شبکه می باشد که در اصطلاح فنی به آن IMSI گفته می شود اگر الان شما موبایلی را که در اختیار دارید خاموش کرده و سیم کارت آن را خارج کنید در قسمت پشت سیم کارت شماره ای حک شده است که 10 رقم از آن به اضافه 43211 (برای شبکه IR-TCI) که مجموعاً 15 رقم می شود را IMSI می نامند البته 10 رقم ذکر شده به تنهایی MSIN نام دارد . در سیم کارتهای جدید 10 رقم آخر شماره ثبت شده در پشت سیم کارت همان MSIN می باشد . یکی از کاربردهای آن در هنگام سوختن ، مفقود شدن و یا دزدیده شدن سیم کارت است اپراتور با همان شماره موبایل قبلی



(MSISDN) سیم کارت جدید با شماره سیم کارت جدید صادر کرده و شماره سیم کارت قبلی شما را از شبکه حذف می کند که اصطلاحاً گفته می شود سیم کار سوزانده شده است .

تذکر : این شماره (IMSI) در حافظه سیم کارت ذخیره شده است .

در شبکه موبایل در هنگام تماس با يك موبایل بیشتر این شماره سیم کارت است که ردو بدل می شود بدین صورت که وقتی ما با شماره موبایلی تماس می گیریم بعد از تایید HLR (محل ثبت دائمی سیم کارت) و گرفتن شماره سیم کارت توسط شبکه ارتباطات بعدی توسط این شماره صورت می گیرد .

دیگر کاربردی که می توان برای آن متصور بود هنگام پیدا کردن شبکه می باشد . وقتی شما با گوشی خود وارد قسمت جستجوی شبکه می شوید بعد از گرفتن امواج (مثلاً در تهران) شبکه موبایلها موجود مثل IR-TCI و IR VALIACOM را مشاهده می کنید و با انتخاب IR-TCI شماره سیم کارت شما از HLR مربوطه استعلام می شود و سپس شما اصطلاحاً REGISTER شده می توانید تماس بگیرید و یا با شما تماس گرفته شود . همین روش در رومینگ می باشد یعنی شما وقتی به عربستان سفر می کنید با جستجوی شبکه عربستان توسط گوشی خود اقدام کرده و نام SA ALJAVAL را انتخاب می کنید از این به بعد با کمک شبکه عربستان و شماره سیم کارت خود در شبکه شناخته شده هستید (البته کشوری که شما به آن سفر می کنید باید با ایران قرارداد رومینگ داشته باشد که تا لحظه نگارش این متن حدود 37 کشور با ایران رومینگ دارند).

:TMSI

این کد شبیه IMSI که همان شماره سیم کارت می باشد هست با این تفاوت که بعد از هر تماس سوئیچ موبایل کدی را به صورت هگزادسیمال ۸ کاراکتری است به صورت تصادفی به شماره سیم کارت مورد نظر نسبت داده و ابتدا با این کد مشترک موبایل را در صورت تماس گرفته شدن خبر می کند البته این کد قبلاً در اختیار گوشی موبایل قرار گرفته و در حافظه گوشی ذخیره می شود (لازم به ذکر است دوبار شبکه با کد TMSI مشترک را پیچ می کند اگر پیدا نشد به فرض اینکه گوشی از این کد خبر ندارد با IMSI پیچ می شود که این کد در حافظه سیم کارت موجود می باشد).

این بخاطر امنیت موبایلی است که به آن زنگ خورده است می باشد چراکه اگر مابه صورت موازی سوئیچ پیش فرضی داشته باشیم و از شماره سیم کارت مشترک نیز مطلع باشیم می توانیم آن را پیچ کنیم ولی با این کد عملاً این مورد غیر ممکن است .

: CI و LAC

در نوشتار های پیشین گفته شد که شبکه موبایل به صورت سلول های 6 ضلعی تقسیم و در نظر گرفته می شود و سایت BTS که غالباً سه جهت (سکتور) دارد بین سه سلول قرار می گیرد و هر جهت يك سلول را پوشش می دهد . در شبکه برای پیدا کردن موقعیت يك مشترک می بایست هر سلول دارای کدی باشد که به این کد (CELL ID) CI گفته می شود که این کد در ایران 5 رقمی می باشد .



حال يك شهر را به چند منطقه بزرگ كه خود اين مناطق شامل چندین CI مي باشد تقسیم مي کنند و به آن LAC مي گویند شهري مانند تهران غالبا به چند LAC و شهرهاي کوچکتر به يك LAC تقسیم مي شوند. کد LAC معمولا 4 رقمي است .

مثال : به طور مثال در میدان ولیعصر تهران يك BTS نصب مي کنیم كه داراي سه جهت (سکتور) مي باشد به اين سلولها مثلا در LAC به شماره 1211 و در CI هاي 12115 و 22115 و 32115 تعريف مي شود و اگر شما تحت پوشش سکتور سوم اين سايت باشيد در VLR اين آدرس ثبت مي شود LAC=1211 و CI=32115 و اين آدرس شما در سويچ مي باشد.

البته لازم به ذکر است كه سويچ موبایل براي پيدا كردن شما هنگام تماس با موبایل تان مجبور است در كل LAC عمل پيچينگ را انجام دهد .

ماژول GSM (SIM 900)





منابع

تاریخچه RFID در سایت ir.vista و تاریخچه GSM در سایت ammepu.blogfa.com

میکروکنترلر 32 بیتی ARM مولف : رضا سپاس یار

تشکر از

استاد دکتر ملک محمد و مهندس شهیدی که در این پروژه زحمات زیادی را متحمل شدند.