



NT 3

PROGRAMACION ON-LINE DE TODA LA FAMILIA APPCON

Introducción

El objetivo de esta nota técnica es que el usuario tenga la capacidad de configurar los parámetros de los módulos de toda la familia APPCON a través de un MCU.

Grupos de la familia según sus características de programación

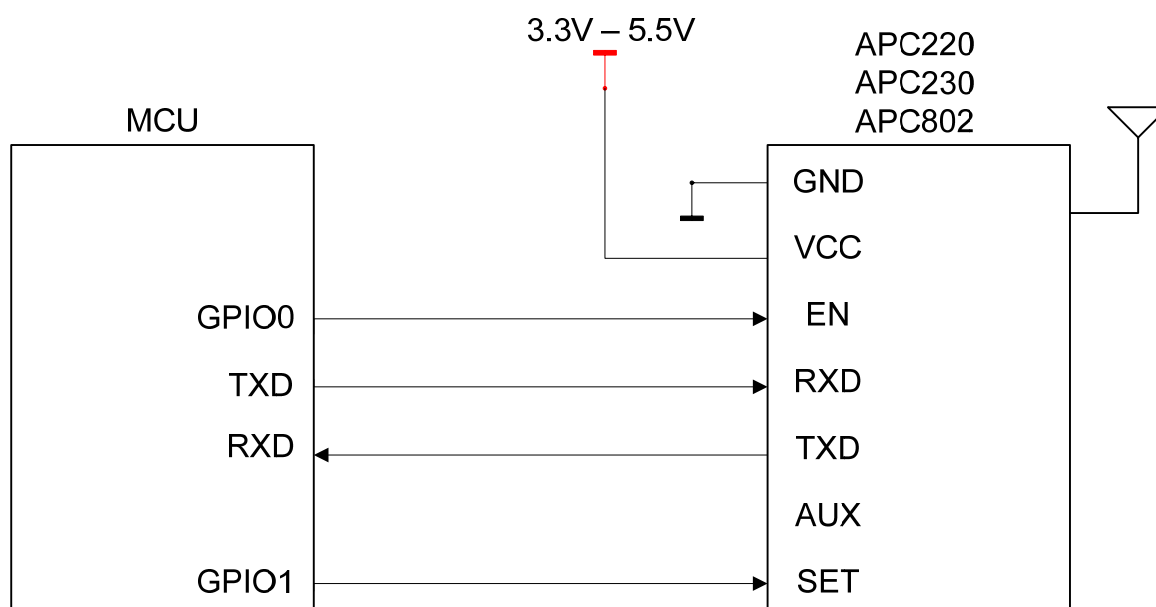
La configuración de parámetros se separan en tres categorías, por lo cual van a existir tres procedimientos diferentes para la programación dependiendo del módulo que estamos utilizando.

Grupos:

1. APC220, APC230, APC802
2. APC200
3. APC240, APC250

Programación del Grupo 1: APC220, APC230, APC802

Para la programación on-line de los parámetros del módulo, debemos primeramente asegurarnos de una correcta conexión eléctrica entre el microcontrolador y el módulo. Esquemáticamente es de la siguiente manera:



PINOUT:

TXD y RXD: Pines de la UART del microcontrolador utilizados para transmitirle información al módulo tanto para transmitir y recibir datos como para cargarle los parámetros de configuración.

EN: Pin de habilitación. Tanto para la configuración como para la transmisión y recepción de datos se debe poner este pin al estado lógico "1".

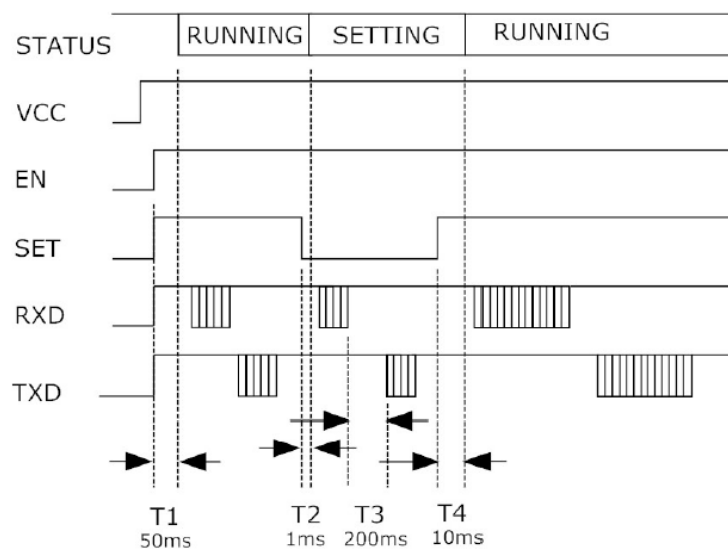
SET: Pin de estado del módulo. El módulo puede colocarse en 2 modos de funcionamiento:

1. **RUNNING:** En este estado el módulo transmitirá los datos recibidos
2. **SETTING:** En este estado el módulo tomará los datos recibidos como parámetros de configuración.

En lo que respecta a esta nota técnica, debemos colocar al módulo en estado SETTING para poder configurar los parámetros deseados.

NIVEL LOGICO DEL PIN SET	FUNCION
'1'	Estado de funcionamiento normal (RUNNING)
'0'	Estado de configuración (SETTING)

Diagrama de tiempos para la correcta configuración de los parámetros:



En este diagrama se puede observar que para enviar comandos de programación debemos poner a '0' el pin de SET y esperar un tiempo T2 mayor a 1ms para comenzar a enviar los comandos de configuración. Cabe aclarar que en el diagrama el pin descrito como RXD es el del módulo, por lo tanto es el PIN TXD del microcontrolador por el cual se transmiten los comandos. El pin TXD del diagrama es el pin RXD del microcontrolador sobre el cual se reciben las respuestas a los comandos emitidos por el pin TXD.

Primeramente debemos tener configurado nuestro MCU con los siguientes parámetros serie:

Interface: UART TTL
Baude Rate: 9600bps
Paridad: Sin paridad

La configuración se realiza a través de código ASCII.

PROTOCOLO PARA CONFIGURACION DE PARÁMETROS

Comando	(32)	Para 1	(32)	Para 2	(32)	Para 3	(32)	Para 4	(32)	Para 5	\r	\n
---------	------	--------	------	--------	------	--------	------	--------	------	--------	----	----

Comando: Son 2 bytes e indica si vamos a escribir o leer datos del módulo.

1. ASCII: WR DECIMAL: (87;82) HEXA : (0x57; 0x52) indica que se van a escribir parámetros en el módulo.
2. ASCII: RD DECIMAL: (82;68) HEXA : (0x52; 0x44) indica que se van a leer los parámetros del módulo.

(32): Es un byte. Es un valor decimal fijo que indica ESPACIO en código ASCII y sirve para separar los parámetros. En Hexadecimal es 0x20.

Para x: Son los distintos valores que le asignamos a cada parámetro. La cantidad de byte y la información que se envía cambia para cada parámetro según la siguiente tabla:

Tabla de parámetros		
Parámetro	bytes	Formato
Frequency (para 1)	6	La unidad es el Khz, por ejemplo 434MHz es 434000
Air rate (para 2)	1	1: 2400 2: 4800 3: 9600 4: 19200
Output power (para 3)	1	0 a 9, 0 expresa -1dBm, 9 expresa 13dBm(20mW)
Series data rate (para 4)	1	0, 1, 2, 3, 4, 5, 6 expresa respectivamente 1200, 2400, 4800, 9600, 19200,38400,57600bps
Series checkout (para 5)	1	0: sin paridad 1: paridad par 2: paridad impar

¡¡¡IMPORTANTE!!!: los parámetros se escriben en código ASCII, o sea que un 0(cero) no es realmente el número cero, sino el valor ASCII que corresponde al cero, el cual es en decimal 48 y en hexadecimal 0x30.

Por ejemplo vamos a setear a un APC220-43 con las siguientes características:

Frecuencia = 434Mhz
Rf data rate = 9600 bps
Output power = 20mW
Serie data rate = 1200 bps
Paridad = sin paridad

El armado de la trama a enviar en código ASCII es el siguiente:

WR_434000_3_9_0_0

En código hexadecimal la trama queda de la siguiente manera:

**0x57,0x52,0x20,0x34,0x33,0x34,0x30,0x30,0x30,0x30,0x20,0x33,0x20,0x39,0x20,0x30,0x20,0x30,
0x0D,0x0A**

La respuesta del modulo en ASCII es la siguiente:

PARA_434000_3_9_0_0

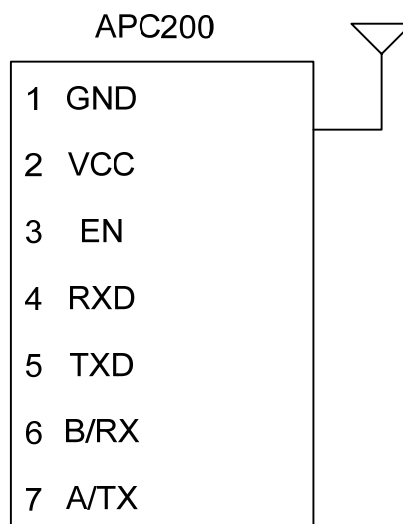
La respuesta del módulo en Hexadecimal es la siguiente:

**0x50,0x41,0x52,0x410x20,0x34,0x33,0x34,0x30,0x30,0x30,0x20,0x33,0x20,0x39,0x20,0x30,
0x20,0x30,0x0D,0x0A**

Programación del Grupo 2: APC200

Este módulo tiene la particularidad de no contar con el pin de SET de los módulos del grupo 1. Esto se debe a que el módulo APC200 tiene interface RS232 y RS485. Utilizando estas funciones en los pines 6 y 7. Veamos el pin out del módulo:

PIN	FUNCION	DESCRIPCION
1	GND	0V
2	VCC	3.3V – 5.5V
3	EN	>1.6V o no conectar
4	RXD	RECEPCION DE UART
5	TXD	TRANSMISION DE UART
6	B/RX	RS485- o RS232 RX
7	A/TX	RS485+ o RS232 TX



A causa de no contar con el pin de set (el cual en los demás grupos indica SETTING o RUNNING) se debe realizar un protocolo de envío de comandos para pasar al módulo al estado de SETTING y una vez que el módulo pasa a este estado, recién en ese momento podemos enviarle la trama de programación de los parámetros.

Pasando al APC200 al estado de SETTING:

Primeramente debemos tener configurado nuestro MCU con los siguientes parámetros serie:

Interface: UART TTL
Baude Rate: 115200bps
Paridad: Sin paridad

Seguir los siguientes pasos:

1. Colocar el PIN EN = 0.
2. Enviar : "0xAA, 0xFF, 0xAA, 0xFF"
3. Habilitar el módulo colocando PIN EN = 1.
4. El módulo debe responder 0xA5, si esto no ocurre luego de 50ms, hay que comenzar nuevamente desde el paso 1. Si esto ocurre seguimos con el paso 5.
5. Detenemos el envío durante 10ms

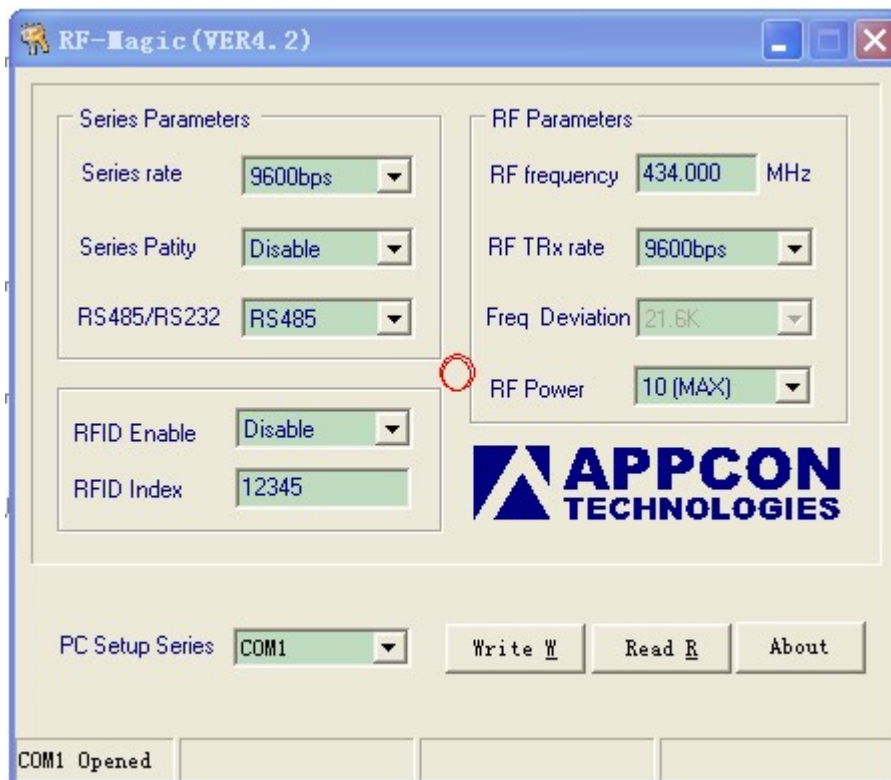
6. En este punto el módulo se encuentra en estado de SETTING por lo cual podemos enviarle la trama de configuración.
7. Enviamos: **0xc3, (36 bytes de configuración), 1 byte (cualquier valor)**. Los 38 bytes enviados no necesitan enviarse con comas de separación.
8. Entonces el módulo responde con los 36 bytes de datos de configuración recibidos. Si no responde con esta cadena se debe comenzar nuevamente por el paso 1.
9. Luego se debe deshabilitar el módulo colocando PIN EN = 0 durante 100ms.

Para verificar que el procedimiento realmente configuró el módulo, se puede colocar al módulo en la interface usb o rs232 y verificar realizando una lectura de parámetros a través de la PC con el RF-MAGIC 4.2.

Que valores colocar en la cadena de 36 bytes de configuración:

Para saber que valores debemos enviarle al módulo para que los parámetros programables tomen el valor deseado por el usuario, lo que debe hacerse es lo siguiente:

1. Se abre el RF-MAGIC 4.2
2. Se colocan a través de los menús desplegables los parámetros con los valores deseados.
3. Se hace doble click en el lugar indicado por el circulo rojo en la siguiente imagen:



4. Esto expandirá la ventana del programa mostrando dos cuadros de texto, uno indicando los datos enviados por el software y otro con los datos recibidos en respuesta del módulo.
5. Una vez que pusimos los valores que deseamos hacemos click en Write W lo cual generará la trama y la mostrará en el cuadro de texto superior. Luego esos 36 bytes generados por el RF-MAGIC son los que debemos utilizar para configurar al módulo con los parámetros seleccionados
6. Este procedimiento lo debemos repetir con las diferentes combinaciones de parámetros que queramos que nuestra aplicación le cambie al APC200.

Programación del Grupo 3: APC240 y APC250

En este grupo, si bien el concepto es el mismo tanto para el APC240 y APC250, vamos a tratarlos de forma independiente ya que tienen pequeñas diferencias.

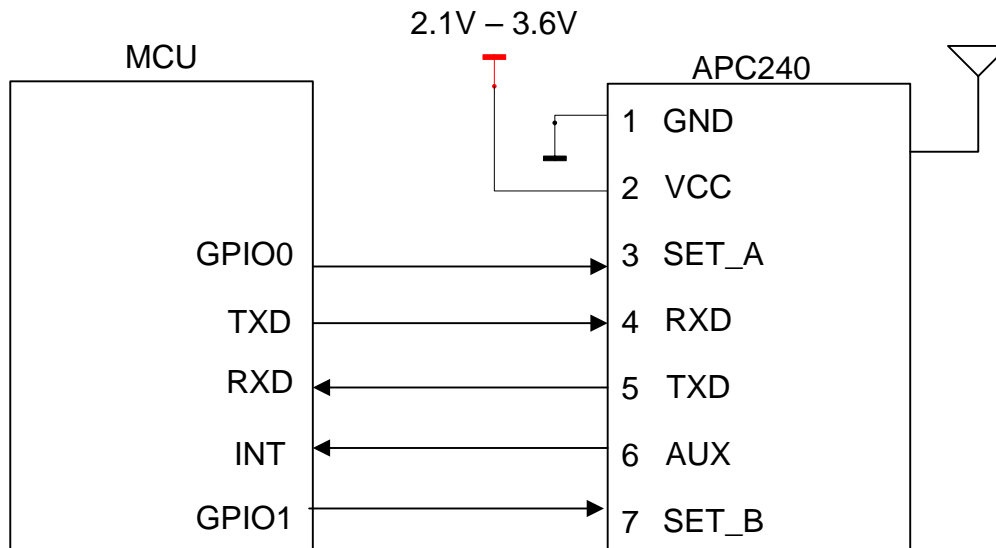
Los módulos de este grupo deben ser puestos en estado de SLEEP para poder ser configurados.

Si bien estos módulos cuentan con pines de SET, los mismos no cumplen la función de pasar a los módulos al estado de SETTING como en el caso del grupo 1.

Vamos a hacer el tratamiento por separado de ambos módulos.

A. Configurando el APC240

CONEXIÓN ELÉCTRICA Y PINOUT:



PIN	NOMBRE	FUNCION	DESCRIPCION
1	GND	GROUND	0V
2	VCC	POWER	2.1V A 3.6V
3	SET_A	INPUT	CAMBIO MODO DE FUNCINAMIENTO EN CONJUNTO CON SET_B
4	RXD	INPUT	UART INPUT TTL
5	TXD	OUTPUT	UART OUTPUT TTL
6	AUX	OUTPUT	SALIDA QUE INDICA SI EL MODULO ESTA RECIBIENDO O TRANSMITIENDO DATOS
7	SET_B	INPUT	CAMBIO MODO DE FUNCINOAMIENTO EN CONJUNTO CON SET_A

Condiciones para programación:

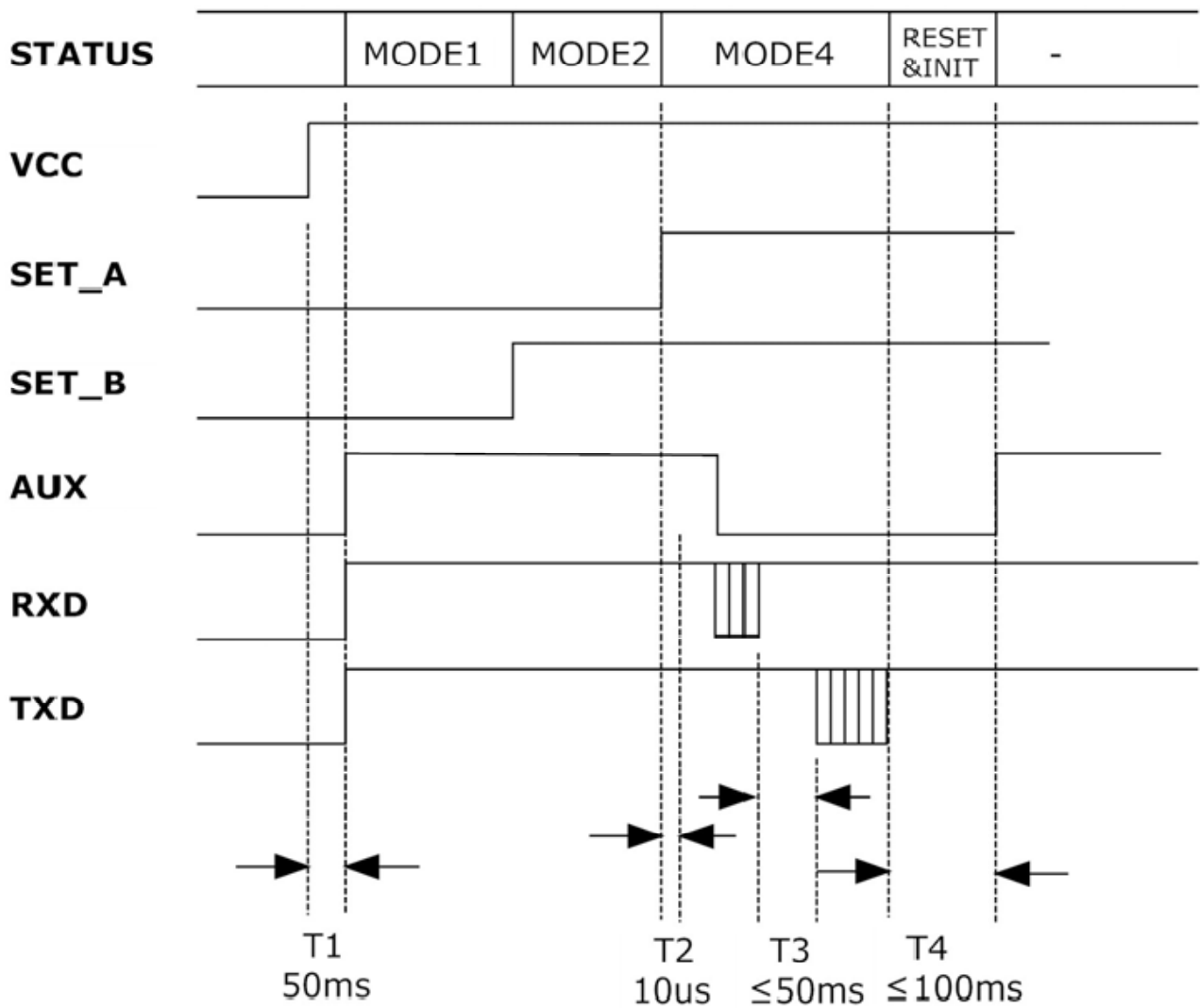
Primeramente debemos tener configurado nuestro MCU con los siguientes parámetros serie:

Interface: UART TTL
Baude Rate: 9600bps
Paridad: Sin paridad

Para poder programar al APC240 debemos primero pasarlo a estado de funcionamiento en MODO 4 (ver hoja de datos para detalles de los modos de funcionamiento del módulo), este modo es el estado de SLEEP. Para ello debemos energizar el módulo, esperar 50ms como mínimo y colocar los pines 3 y 4 en estado '1', este es el MODO 4.

Una vez que el módulo pasa a este estado debemos verificar que el pin AUX se encuentra en estado alto '1' lo cual indica que el módulo no se encuentra ocupado. Una vez corroborado esto, esperamos como mínimo 10us y podemos pasar a enviarle la trama de configuración al módulo a través de RXD.

Diagrama de tiempos:



Parámetros y protocolo:

Los parámetros programables son los siguientes: [\(para detalles de que indica cada parámetro en particular por favor referirse a la hoja de datos de módulo APC240\)](#)

Parámetro	Unidad	Longitud (byte)	Valores
Freq.	Khz	3	433920 = 0x06,0x9f,0x00
DR_FSK	Kbps	1	1,2,3,10,20,40 equivale a 0x00,0x01,0x02,0x03,0x04,0x05
Pout	dB	1	Desde 0 a 7 representado por 0x00 a 0x07.
DR_IN	Kbps	1	1.2, 2.4, 4.8, 19.2, 38.4, 57.6 equivale a 0x00,0x01,0x02,0x03,0x04,0x05,0x06
Paridad	--	1	0x00: sin paridad 0x01: Paridad par 0x02: Paridad impar
Tw	Seg.	1	0.05, 0.1, 0.2, 0.4, 0.6, 1, 1.5, 2, 2.5, 3, 3.5, 4, 5 correspondiente a 0x00 al 0x0b.

Para programar al dispositivo hay dos comandos: Comando de lectura y comando de escritura.

Comando de Lectura: 0xff, 0x56, 0xae, 0x35, 0xa9, 0x55, 0xf0

Respuesta del módulo: 0x24, 0x24, 0x24 + Freq + DR_FSK + Pout + DR_IN + Paridad + Tw

Comando de Escritura: 0xff, 0x56, 0xae, 0x35, 0xa9, 0x55, 0x90 + Freq + DR_FSK + Pout + DR_IN + Paridad + Tw

Respuesta del módulo: 0x24, 0x24, 0x24 + Freq + DR_FSK + Pout + DR_IN + Paridad + Tw

Ej: si queremos programar un APC240 con los siguientes parámetros:

- Freq = 433.92Mhz
- DR_FSK = 10Kbps
- Pout = 10dbm
- DR_IN = 9.6Kbps
- Paridad = Sin paridad
- Tw = 1 segundo

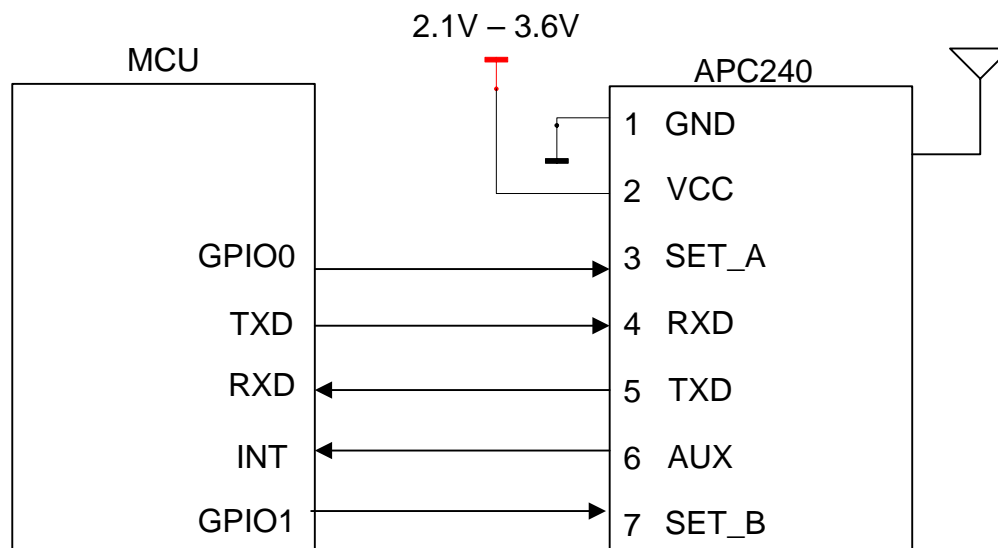
Entonces debemos generar la siguiente trama:

Escritura: 0xff,0x56,0xae,0x35,0xa9,0x55,0x90,0x06,0x9f,0x00,0x03,0x07,0x03,0x00,0x05

Respuesta: 0x24,0x24,0x24,0x06,0x9f,0x00,0x03,0x07,0x03,0x00,0x05

B. Configurando el APC250

CONEXIÓN ELÉCTRICA Y PINOUT:



PIN	NOMBRE	FUNCION	DESCRIPCION
1	GND	GROUND	0V
2	VCC	POWER	2.1V A 3.6V
3	/EN	INPUT	PIN DE HABILITACION. '0' NORMAL, '1' SLEEP
4	RXD	INPUT	UART INPUT TTL
5	TXD	OUTPUT	UART OUTPUT TTL
6	AUX	OUTPUT	SALIDA QUE INDICA SI EL MODULO ESTA RECIBIENDO O TRANSMITIENDO DATOS
7	SET	INPUT	INDICA AL MÓDULO SI MOSTRAR LA INFORMACION QUE RECIBE O LA INTENSIDAD DEL CAMPO RECIBIDO

Condiciones para programación:

Primeramente debemos tener configurado nuestro MCU con los siguientes parámetros serie:

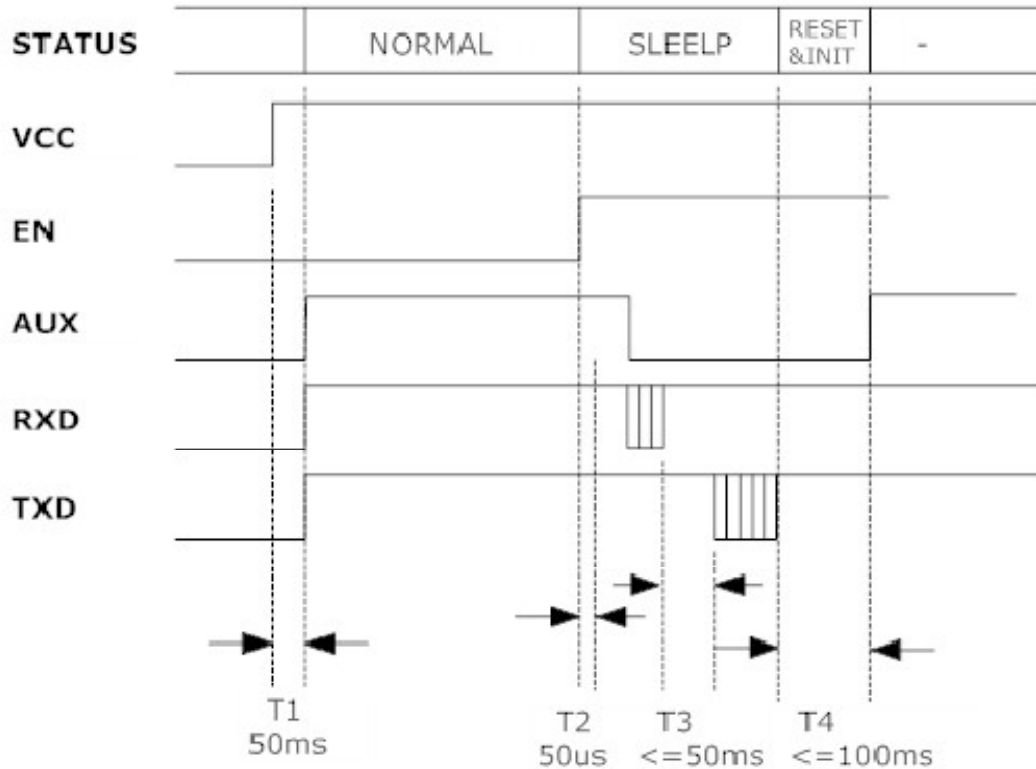
Interface: UART TTL

Baude Rate: 9600bps

Paridad: Sin paridad

Para poder programar al APC245 debemos primero pasarlo al estado de SLEEP. Para ello debemos colocar el pin EN = 1. Y luego debemos monitorear el pin AUX. Cuando AUX se pone a '1' indica que el módulo no está ocupado, entonces puede recibir los comandos de programación (esto luego de 20us como mínimo).

Diagrama de tiempos:



Parámetros y protocolo:

Los parámetros programables son los siguientes: [\(para detalles de que indica cada parámetro en particular por favor referirse a la hoja de datos de módulo APC250\)](#)

Parámetro	Unidad	Longitud (byte)	Valores
Freq.	Khz	3	433920 = 0x06,0x9f,0x00
DR_FSK	Kbps	1	1,2,3,10,20,40 equivale a 0x00,0x01,0x02,0x03,0x04,0x05
Pout	dB	1	Desde 0 a 7 representado por 0x00 a 0x07.
DR_IN	Kbps	1	1.2, 2.4, 4.8, 19.2, 38.4, 57.6 equivale a 0x00,0x01,0x02,0x03,0x04,0x05,0x06
Paridad	--	1	0x00: sin paridad 0x01: Paridad par 0x02: Paridad impar

Para programar al dispositivo hay dos comandos: Comando de lectura y comando de escritura.

Comando de Lectura: 0xff, 0x56, 0xae, 0x35, 0xa9, 0x55, 0xf0

Respuesta del módulo: 0x24, 0x24, 0x24 + Freq + DR_FSK + Pout + DR_IN + Paridad

Comando de Escritura: 0xff, 0x56, 0xae, 0x35, 0xa9, 0x55, 0x90+ Freq + DR_FSK + Pout + DR_IN + Paridad

Respuesta del módulo: 0x24, 0x24, 0x24 + Freq + DR_FSK + Pout + DR_IN + Paridad

Ej: si queremos programar un APC250 con los siguientes parámetros:

- *Freq = 433.92Mhz*
- *DR_FSK = 10Kbps*
- *Pout = 10dbm*
- *DR_IN = 9.6Kbps*
- *Paridad = Sin paridad*

Entonces debemos generar la siguiente trama:

Escritura: 0xff,0x56,0xae,0x35,0xa9,0x90,0x06,0x55,0x9f,0x00,0x03,0x07,0x03,0x00

Respuesta: 0x24,0x24,0x24,0x06,0x9f,0x00,0x03,0x07,0x03,0x00



CTM Electrónica
J. M. Bustillo 3279
(C1406HJA) C.A.B.A.
Argentina

Tel./Fax: +54 (11) 4619 1370
www.ctmelectronica.com.ar