

GPS Tracker

Simón Bruno

Facundo Cárdenas

Introducción

Motivación:

- Alarmas de autos que no paran de sonar
- Autos robados
- Gran diversidad de aplicación

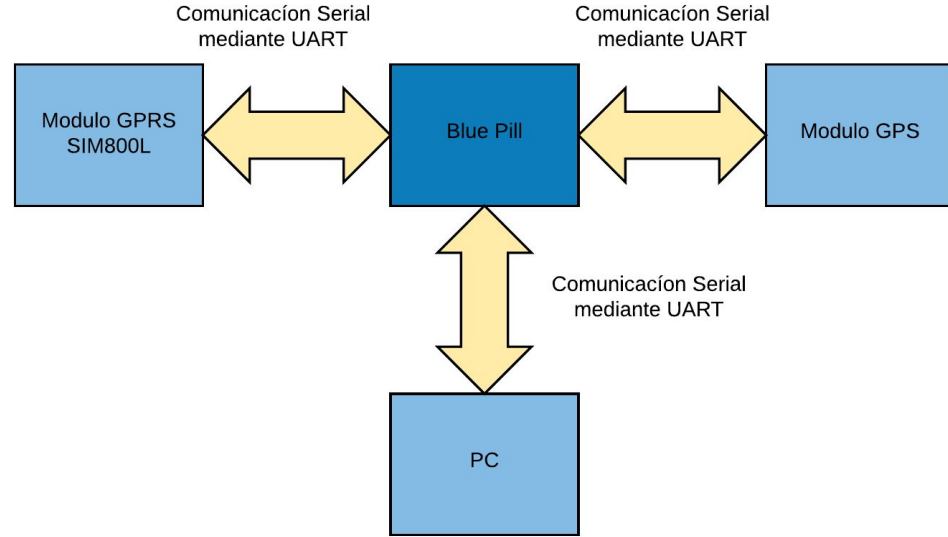
Propuesta:

- Rastreo de posición a través de SMS

Esquema Conceptual y Tecnológico



Diagrama de Bloques



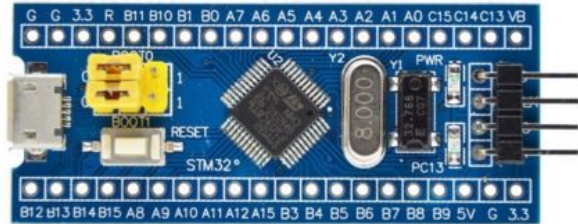
Módulo GSM/GPRS SIM800L



Módulo GPS Ublox Gy-neo6mv2



Microcontrolador STM32F103C8T6 - BluePill



Computadora

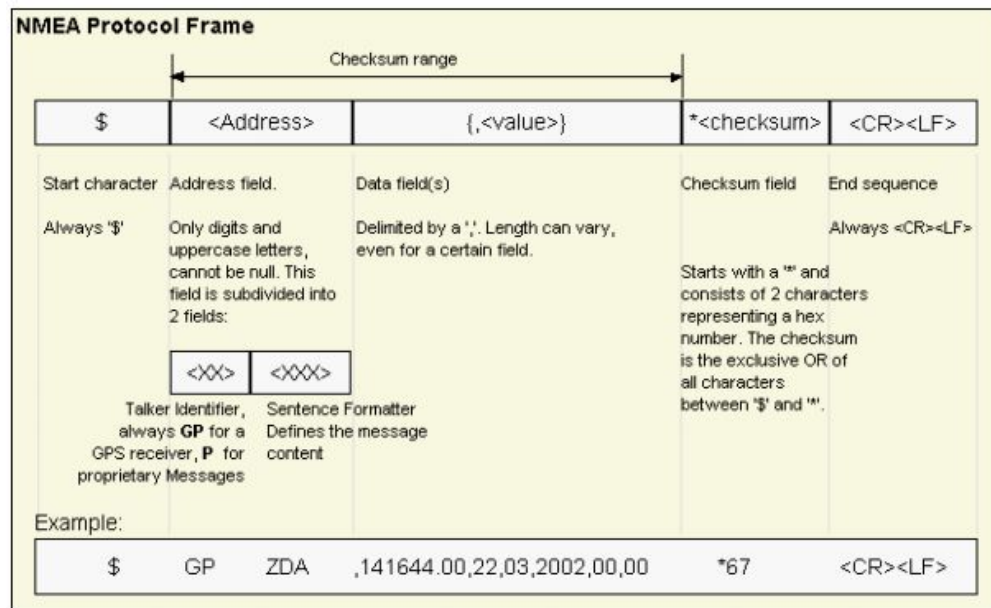


Módulos utilizados

Funcionamiento GPS

Del módulo GPS recibimos oraciones formateadas según el protocolo NMEA (National Marine Electronics Association), que brindan información sobre posición, velocidad, satélites, calidad de la señal, tiempo, etc.

Para el proyecto utilizamos los mensajes de propietario PUBX para obtener posición y hora manteniendo al módulo en stand-by.



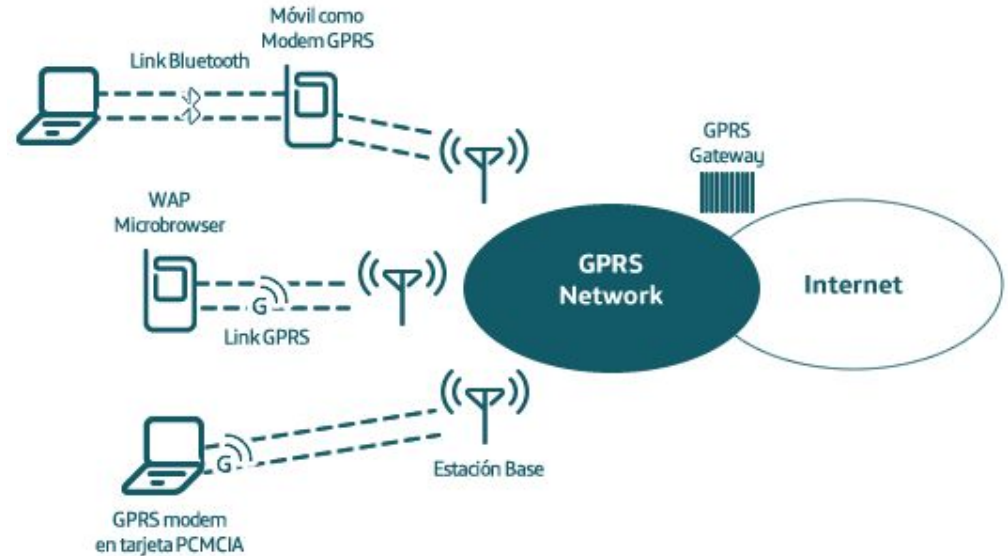
Message Structure:

```
$PUBX,00,hhmmss.ss,Latitude,N,Longitude,E,AltRef,NavStat,Hacc,Vacc,SOG,COG,Vvel,ageC,HDOP,VDOP,TDOP,
,GU,RU,DR,*cs<CR><LF>
```

Funcionamiento GSM/GPRS

A través de comandos AT enviados por UART, podemos configurar y determinar las tareas a realizar por el SIM800L.

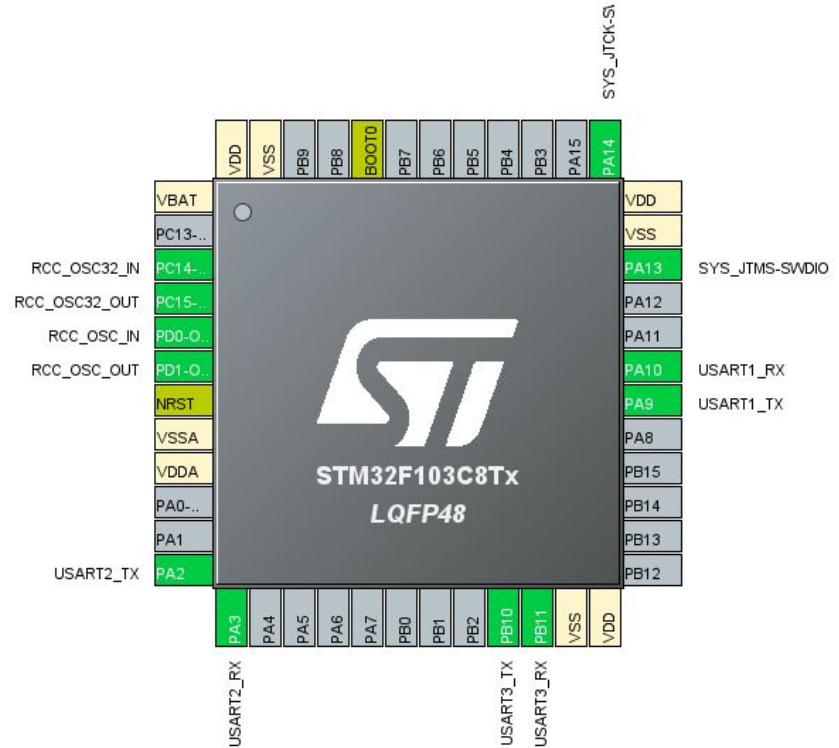
AT command	Meaning
+CMGS	Send message
+CMSS	Send message from storage
+CMGW	Write message to memory
+CMGD	Delete message
+CMGC	Send command
+CMMS	More messages to send



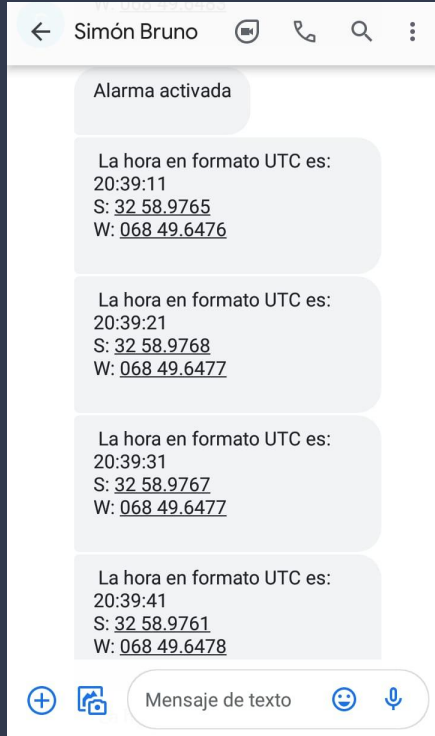
Funcionamiento Bluepill

Configuramos el microcontrolador utilizando el software STM23CubeIDE.

- 3 puertos serie UART para la comunicación con los 2 módulos y la PC.
- 1 timer para el tiempo de descanso entre mensajes.

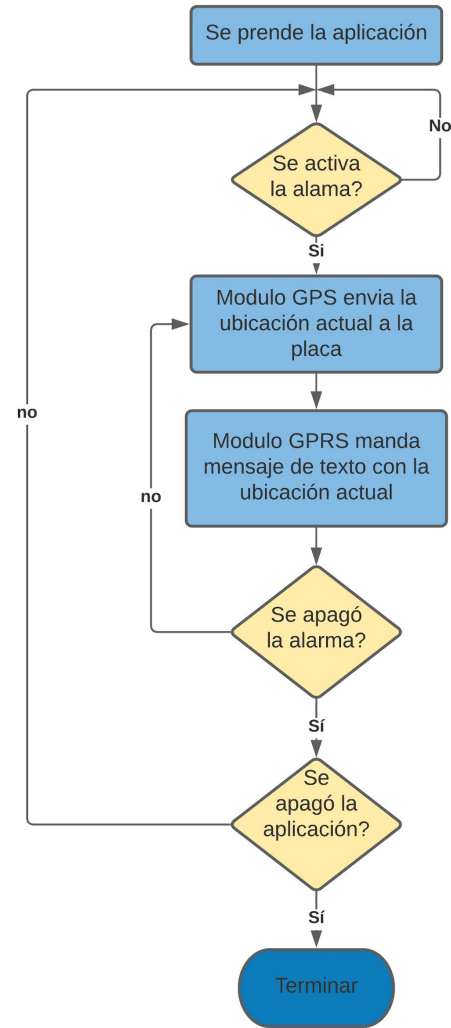


Funcionamiento general



Interfaz usuario

Diagrama de flujo



Funcionamiento del programa

```
423 /* USER CODE BEGIN 4 */
424 void text_msn(mensaje){
444
445 void interpreta_gps(void){
525 void interpreta(void)
584 void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
615
616 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
620
621 /* USER CODE END 4 */
```

El proyecto está condensado en esas funciones y rutinas de interrupción del microcontrolador.

La idea general es recibir los datos de posición y hora, unirlos en un mensaje de texto, y enviarlo al celular. Las tareas principales realizadas son de manipulación y coordinación de mensajes.

La función *interpreta* también contiene los protocolos para los comandos por computadora de configuración del timer o la activación de la alarma.

Obtención de la posición

```
452 for (char *pV = strtok(buff, ","); pV != NULL; pV = strtok(NULL, ",")) {
453     switch (cnt) {
454         case 2:
455             utcRaw = strdup(pV);
456             break;
457         case 3:
458             latRaw = strdup(pV);
459             break;
460         case 4:
461             hemNS = strdup(pV);
462             break;
463         case 5:
464             lonRaw = strdup(pV);
465             break;
466         case 6:
467             hemEW = strdup(pV);
468             break;
469     }
470     cnt++;
471 }
```

Primero recibimos el mensaje del GPS, que contiene más información de la que necesitamos, por lo que debemos depurar los datos y quedarnos con lo útil.

Haciendo uso de la función *strtok*, separamos en tokens el mensaje recibido del GPS y guardamos cada uno de los argumentos en una variable distinta según corresponda.

Envío de mensaje de texto

```
424 void text_msn(mensaje){
425     HAL_UART_Transmit(&huart1,(uint8_t *)AT,strlen(AT), HAL_MAX_DELAY);//Probamos el modulo
426     HAL_UART_Transmit(&huart2,(uint8_t *)AT,strlen(AT), HAL_MAX_DELAY);//Probamos el modulo
427
428
429     HAL_UART_Transmit(&huart1, (uint8_t *)ModoM,strlen(ModoM), HAL_MAX_DELAY);//Ponemos el modulo en modo mensaje
430     HAL_UART_Transmit(&huart2, (uint8_t *)ModoM,strlen(ModoM), HAL_MAX_DELAY);//Ponemos el modulo en modo mensaje
431
432
433     HAL_UART_Transmit(&huart1, (uint8_t *)Number,strlen(Number), HAL_MAX_DELAY);//numero al que le mandamos mensaje
434     HAL_UART_Transmit(&huart2, (uint8_t *)Number,strlen(Number), HAL_MAX_DELAY);//numero al que le mandamos mensaje
435
436
437     HAL_UART_Transmit(&huart1, (uint8_t *)mensaje,strlen(mensaje), HAL_MAX_DELAY);//mensaje que mandamos
438     HAL_UART_Transmit(&huart2, (uint8_t *)mensaje,strlen(mensaje), HAL_MAX_DELAY);//mensaje que mandamos
439
440
441     HAL_UART_Transmit(&huart2, &ctrlZin, 1, 100);
442 }
```

Después de obtener los datos y de armar el mensaje a enviar, seteamos el SIM800L enviando los comandos AT correspondientes.

Establecemos conexión, seleccionamos el Modo Mensaje, introducimos el numero de telefono y proveemos el mensaje.

Resultados

- Comunicación satisfactoria entre los módulos.
- Fuerte dependencia de la calidad de la señal de los satélites de GPS y antenas de teléfono.
- Implementación de los conocimientos adquiridos durante el cursado.
- Solución de problemas de manera propia y basada en investigación.

Conclusiones

- Aprendizaje sobre el uso de un microcontrolador potente y conveniente.
- Aplicación útil y cotidiana.
- Gran variedad de alternativas de funcionamiento.
- Buen prospecto de mejoras en la implementación.
- Muchas cosas para aprender y seguir creciendo como futuros ingenieros.

Muchas gracias.