Telit AppZone Programming Tips

Table of Contents

Basic Operations

1. Print messages to UART

In order to print messeges to the UART use 1 of the 2 functions:

- `void PrintToUart(const char *fmt, ...);`
- `M2M_T_HW_UART_RESULT m2m_hw_uart_write (M2M_T_HW_UART_HANDLE handle, char *buffer, int len, int *len_sent);`

Code example:

```
void M2M_main(void)
{
PrintToUart("hello world");
}
```

```
void M2M_main(void)
{
int sent;
M2M_T_HW_UART_HANDLE local_fd;
local_fd = m2m_hw_uart_open();

  if (M2M_HW_UART_HANDLE_INVALID != local_fd)
  {
       m2m_hw_uart_write(local_fd, "hello world" , strlen("hello world"), &sent);
       m2m_hw_uart_close(local_fd);
  }
}
```

*** extern void PrintToUart(const char *fmt, ...); should be written in the functions prototypes section.**

2. Working with processes

In order to start a process work, use the following function:

- **void m2m_os_send_message_to_task**(**short** task_number, **int** type, **long** param1, **long** param2);

Where the task_number is the number of the process and the other parameters represents the actions to be performed at the process.
This function invokes the function:

- **int** M2M_msgProc1(**int** type, **long** param1, **long** param2)

That is in M2M_proc1.

Code example:

```
void M2M_main(void)
{
m2m_os_send_message_to_task (1, 0, 0, 0);
}

int M2M_msgProc1(int type, long param1, long param2)
{
  // the code to be executed in process 1
}
```

3. File System

The Telit AppZone allows you to get access to the module's file system and execute different operations on its files.
There are several functions that can be used in order to work with the FS:

- `int m2m_fs_clear(void);`
- `M2M_T_FS_HANDLE m2m_fs_open(char *filename, int mode);`
- `int m2m_fs_close(M2M_T_FS_HANDLE filehandle);`

and many more.

Code example:

```
void M2M_main(void)
{
char FileName[20] = "An example file.txt";

m2m_fs_clear();//Deletes all files from M2M file system
PrintToUart("I'm running help function "Create_file()" now.");
m2m_os_sleep_ms(2000);

Create_file(FileName);
/*Help function that include:
1)m2m_fs_create
2)m2m_fs_write
3)m2m_fs_close
*/
}

void Create_file (char *filename)
{
M2M_T_FS_HANDLE file_handle = NULL; //Create handle by name file_handle.
char buf[] = "Hello world!"; //Text that will be written in the created file.

 if(M2M_FS_SUCCESS == m2m_fs_create(filename)) //Checks if file was created properly
 {
 file_handle = m2m_fs_open(filename, M2M_FS_OPEN_APPEND);// On failure m2m_fs_open will return NULL.
   if(NULL != file_handle)
   {
   m2m_fs_write(file_handle, buf, sizeof(buf)); // Write "Hello world!" to the file.
   m2m_fs_close(file_handle); // Close the file.
   }
   else
   {
    PrintToUart("Warning:For some reason the file couldn't be open.");
   }
 }
 else
 PrintToUart("Warning:File wasn't created properly.");
}
```

*** It is a good practice to put the process to sleep for a few seconds between the functions calls, using the function:**

- `void m2m_os_sleep_ms(unsigned long ms);`

4. Timers

The Telit AppZone supports both software and hardware timer. There are 2 HW timers to customer and up to 30 SW timers available for use.

SW Timer

In order to work with sw timer, use the following functions:

- `M2M_T_TIMER_HANDLE  m2m_timer_create(M2M_T_TIMER_TIMEOUT cb, void *arg);`
- `void m2m_timer_start(M2M_T_TIMER_HANDLE timer, unsigned long msecs);`

To create the timer and start its work.

Code example:

```
M2M_T_TIMER_HANDLE user_timer;

void M2M_main(void)
{

    user_timer = m2m_timer_create(timer_handler_function, NULL);
    m2m_timer_start(user_timer,5000); //start the timer after 5 seconds

}

void timer_handler_function(void *arg)
{
    PrintToUart("hello"); // hello will be printed to the UART every second
    m2m_timer_start(user_timer,1000); // recall the timer every second
}
```

HW Timer

In order to work with hw timer use the following function:

- `int m2m_hw_timer1_start(unsigned int span);`

On timeout, the G30 will call the onHWTimer1() callback function.

Code example:

```
void M2M_main(void)
{
       m2m_hw_timer1_start(1000); // will call the M2M_onHWTimer1 function
}

void M2M_onHWTimer1(void)
{
   // the code to be executed in timer 1
}
```

5. SMS

To send SMS message, use the function:

- **int m2m_sms_send_SMS(char* address, char* message);**

Code example:

```
int M2M_msgProc1(int type, long param1, long param2)
{

	int smsResult;
	//Touch unused variables to avoid warnings
	(void)type;
	(void)param1;
	(void)param2;


	m2m_sms_set_text_mode_format();
	PrintToUart("SET TEXT FORMAT");

	m2m_os_sleep_ms(5000);

	PrintToUart("Sending Message: Hello World!!  To: +9725461693045");
	smsResult = m2m_sms_send_SMS("+9725461693045", "Hello World!!");

	m2m_os_sleep_ms(5000);

    if (1 == smsResult) // success
    {
	PrintToUart("SMS WAS SENT");
    }
    else
    {
	PrintToUart("SMS WAS NOT SENT");
    }



    return 0;

}
```

*** It is a good practice give the application couple of seconds delay, using timer, before accessing process 1.**

Code example:

```
void init_timeout_handler(void *arg)
{
	PrintToUart("Start SMS Application");
	m2m_os_send_message_to_task(1,0,0,0);
	(void) arg;
}

void M2M_main(void)
{
	user_timer = m2m_timer_create(init_timeout_handler, NULL);
	m2m_timer_start(user_timer, 10000);
}
```

6. AT Commands

To send AT commands to your module, use the function:

- `int m2m_os_iat_send_at_command(char *atCmd);`

Code example:

```
void SendATE1()
{

   int atResult;
   \\ sends ate1 command to the module
   atResult = m2m_os_iat_send_at_command("ate1\r\n");

        if (1 == atResult) // success
        {
               PrintToUart("AT command WAS SENT");
        }
        else
        {
            PrintToUart("AT command WAS NOT SENT");
        }
}
```

*** make sure to add the "\r\n" suffix to the at command string you are sending.**

7. PDP Context

The PDP related functions are:

- `int  m2m_pdp_activate(char *apn, char *name, char *pwd); \\ Activates a PDP context.`
- `int  m2m_pdp_deactive(void); \\ Deactivates a PDP context.`
- `int m2m_pdp_get_status(void); \\ Gets the status of the PDP connection`
- `unsigned long m2m_pdp_get_my_ip(void); \\ Gets the IP addr of the PDP connection.`

Code example:

```
void Activate_PDP()
{
       int res = 0;

       PrintToUart("START ACTIVATE PDP");
       m2m_os_sleep_ms(3000);


       if (M2M_PDP_STATE_ACTIVE != m2m_pdp_get_status()){
           res = m2m_pdp_activate(GPRS_APN, NULL, NULL);
           //PrintToUart ("PDP ACTIVATE res=%d",res);

           if (M2M_PDP_STATE_FAILURE == res) {
                   PrintToUart ("PDP ACTIVATE FAIL");
           }
       }
       else{
              PrintToUart ("PDP IS ALREADY ACTIVATE ");
       }
    return;
}
```