

12.2.5 Making "Hello, World!"

The process of modifying an application, then downloading and running it on the modem, can be demonstrated by altering the Turnkey application to print "Hello, world!" to the Logger port.

1. In the IAR workspace window, expand the Turnkey group by clicking on the '+' and then double-clicking on APL.c. This will open the file in the editor window:



Figure 12-17: Turnkey APL.c file

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.



- 2. Use Ctrl/F to search for POWER_ON. In the function processEvent(), you should see the line case POWER_ON:
- 3. Modify APL.c by adding printf("\r\nHello, world!\r\n"); under the POWER_ON line. The file with the modifications should appear as in Figure 12-18 below:



Figure 12-18: Modify Turnkey APL.c by adding a printf statement

4. After making the modification, select Make (F7) to save the changes and rebuild the project. If there are errors shown in the bottom window, correct these and rebuild. This creates the new binary file, TurnKey.bin, which you then load into the modem with the QUAKE Configuration Tool (QCT), as described in <u>Section 12.2.2</u>.

Note:

If additional changes are required after compiling, it is a good idea to first select "Clean" from the Project menu before recompiling. This removes the existing object files and insures that the subsequent "Make" does a completely new compile and build, without using any previously compiled object files.

Page 81

res se nt ger "e/r " TIM tur ddi ke hei in <u>c</u> cr c



5. Once you have downloaded the new application file with "Hello, World," cycle power on the modem. You should see the following output on the Logger port:



Figure 12-19: Logger output with "Hello, World!"

12.2.6 Additional detail on Turnkey

The Turnkey application allows the user to set configuration parameters to control when it sends a report and what is contained in the report data. The application can report a GPS position fix, DIO and Analog values, as well as CAN data. It accepts parameter changes received over the air or from a serial port. It can also accept an application update over GPRS. The application can be used as is, or it can serve as the basis for a more complex user application. We have seen how to set up the IDE, and build, load and execute the Turnkey application. We now examine the Turnkey code in more detail in order to understand the QUAKE Event Framework.

S Note:

Turnkey is written in network-agnostic code; that is, the same code can be used for any satellite network with no changes required. Please note that Turnkey is the only sample application written in this manner. All the other sample applications are written in network-dependent code, meaning they must be changed when switching between satellite networks.

The main loop of Turnkey initializes configuration parameters, sets up serial port handling by registering a callback function for each serial port in use, generates a power-on event and enters a **while** loop where it checks for messages in its queue and processes them. Most events are

Document Number 1135-4713 Rev G

Page 82

Page 83



handled in a single function called processEvent(). GPS, serial data, CAN events and some others are handled directly from the main loop. This is shown in the figure below.

.c User	_libQuake.h APL.c
2024	//APL_setSerPortMode(UARTPORT_LOGGER, PORT_MODE_PROTOCOL, 0, 0, 0);
2025	
2026	// Enter the main task loop
2027	while (1)
2028	(
2029	if(SYS_msgQReceive(aplFromQosMsgQId, (u8*)&qMsg, sizeof(qMsg), TICKS_PER_SEC) == 0K)
2030	(
2031	if(qMsg.type != APL_EVENT_MSG && QLM_getLogDownlink())
2032	{
2033	printf("TK: %s: Msg %d rcvd\r\n",FUNCTION, qMsg.type);
2034	}
2035	switch (qMsg.type)
2036	{
2037	case APL_EVENT_MSG:
2038	processEvent(&qMsg.eventOption);
2039	
2040	// Release the memory if it has been allocated
2041	if (dMsg.eventUption.msg != NULL)
2042	{
2043	<pre>iree(dmsg.eventUption.msg);</pre>
2044	}
2045	preak;
2040	CORE ADI MENTI MEC.
2047	distance and the second s
2040	http://www.angle.com/ang
2049	DECAR,
2050	CASE ADI HTTI MSC.
2051	<pre>rrocesultil_hote rrocesultilModeCmd(adMsg utilOntion) .</pre>
2052	hreak.
2054	NA CARY

Figure 12-20: Turnkey main task loop

We have seen in the Turnkey application how processEvent () handles a power-on sequence. Depending on the settings of the configuration parameters, it initializes the satellite and GPRS modules, starts the GPS and sets timers to check for GPS and to transmit message timeouts. There is a single TIMER event which occurs when any of the timers has a timeout. Depending on the timer that expired, Turnkey may build and send a message, or it may check for a received message (from GPRS) or process a timeout from the GPS process.

Other events are triggered, for example, when:

- DTR changes state
- a satellite or GPRS goes in or out of view
- a message is received.

The complete list of events is discussed in Chapter 14.



12.3 QuickStart application

The QuickStart application provides the minimum framework to build a custom application on the Q4000/QPRO. It starts the main application task and lists the events used by the foundation to signal actions to the application.

- 1. Close any APL.c file that is still open before opening the QuickStart APL.c, in order to avoid the confusion of having two APL.c files open at the same time.
- Select the QuickStart Workspace from the drop-down list at the top left-hand corner of the IDE screen. This will cause the Turnkey group to be grayed out, and the QuickStart group will become bright yellow.

	KIAR Embedded Workbench IDE	and the second	_ 0 X
	File Edit View Project Simulator Tools Wind	dow Help	
		- イ > > 注 図 > +> 48 許 😳 % 免	
/	Workspace ×		* >
	QuickStart 🗸		
	Files 🗧 🕅		
CONFIDENTIAL	APL_Orbcomm - QuickStart APL APL DemoAppADC DemoAppFS DemoAppREMOTE DemoAppRTOS DemoAppRTOS DemoAppRTOP Mitachi_zosen APL.c DemoApLc DemoAppCAN Output		

Figure 12-21: Selecting the QuickStart Workspace

3. Open APL.c from the QuckStart application.

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL CORPORATION.



As seen in Figure 12-22, the QuickStart APL.c contains little code under most of the event cases in the switch statement in processEvent(). This allows the user to fill in the event cases with customized code.

🔀 IAR Embedded Workbench IDE				
<u>Eile E</u> dit <u>V</u> iew <u>P</u> roject <u>S</u> imulator <u>T</u> ools <u>W</u> i	indow	Help		
0 🖆 🖬 🕼 🞒 🖓 🐘 🛍 🗠 🗠	cr	eates	GendTKMsa	💌 🛷 🌾 🚾 🛐 🖻 🐡 🏟 跡 🗟 🥨 隊 🧶 🕭 🕭
Workspace		×	APL.c	* x
QuickStart		-	118	
Files	<u>e</u> -	23	119	case NO EVENT:
	(m)	10	120	// The NO_EVENT Event will occur periodically (once per Second by de.
Application - QuickStart	~		121	<pre>// can be changed using the APL_taskSetNoEventInterval() API functio;</pre>
			122	// other event has occurred in that time. Here is where it is recom
			123	// applications perform any periodic status updates that may be requ.
			124	
			125	// Check if any Software Timers have expired.
			126	UIL_checklimers();
		1	127	hrazk
			120	WIEdk,
			130	case TIMER:
			131	break:
			132	
			133	case CAN MSG:
			134	break;
			135	
			136	case ORB_ANTENNA_VSWR:
			137	break;
			138	
			139	case POSITION_FIX:
			140	break;
			141	DOUTITON ALADY.
			142	case POSITION_ALARM:
			143	Break;
			145	case MTS DTP: // MTS Port DTP line has changed state
			146	break:
			147	
			148	case SAT IN VIEW:
			149	break;
			150	
			151	case CELL_NET_IN_VIEW:
			152	break;
			153	
			154	case GLSS_AVAILABLE:
			155	break;
			156	W//C A CV.
			157	Case mot AUK:
			150	predk;
			160	case USER CMD.
			161	break:
Application			161 162	
Application				

Figure 12-22: QuickStart switch statement

The instructions for building, loading and executing the code are the same as in <u>Section 12</u>, except that after building the application, the executable bin file is: .../QuickStart/exe/QuickStart.bin.

Since QuickStart is simply a template and is included as a starting point for a user's custom application, no examples of running the QuickStart application will be shown.

12.4 DemoApp applications

To run the DemoApps, close any open APL.c files and select the DemoAppXXX application on the IAR IDE. Open APL.c under DemoAppXXX. The applications are described in the following sections.

Document Number 1135-4713 Rev G

Page 85

Information classified Confidential - Do not copy (See last page for obligations)



12.4.1 DemoAppGSM

CONFIDENTIAL

DemoAppGSM sets a timer to first send out a short message via GSM and then sets a timer to send out a short report via satellite. It uses network-specific calls to do this. It continues to alternate between GSM and satellite messages until the application is stopped. The timers are hardcoded to 60 seconds and are not adjustable from any program parameters.

1. Select the DemoAppGSM Workspace from the drop-down list at the top, left-hand corner of the IAR IDE screen. Open the APL.c file, as shown below:



Figure 12-23: DemoAppGSM - Selecting the Workspace

2. Now build, load and execute DemoAppGSM. The instructions for building, loading and executing the code are the same as in <u>Section 12</u>, except that after building the application, the executable bin file is: .../DemoAppGSM/exe/xxx-DemoAppGSM.bin.

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

Information classified Confidential - Do not copy (See last page for obligations)



CONFIDENTIAL

3. After startup, check the Logger output for the line **APL DEMO: GSM then Orbcomm.** This indicates that the correct DemoApp is running.



Figure 12-24: DemoAppGSM | Logger output that DemoAppGSM is running

Document Number 1135-4713 Rev G

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.



In the POWER_ON, case statement, the application calls <code>TIMER_setDuration()</code> and sets it to expire after GSM_DEMO_GSM_TIMEOUT_SECS. This causes a TIMER event to occur when the timer expires.

	X
Window Help	
- 🗸 🏷 🐂 🎦 🔯 🗠 🥔 🕼 😳 👷 🕭 🕭	
APL.c	•
<pre>// Also turn on and initialize the GSM module if(NET_isConfigured(NETWORK_TERR)) { if (TERR_powerOn() == ERROR) printf ("TERR_powerOn returned ERROR!\r\n"); }</pre>	1.1
<pre>// Set a timer for 1 minute; if the GSM network does not come into // view in a minute, we'll send via ORBCOMM; if the GSM net does // come into view, we'll reset the timer, and allow 1 minute for // the message to be sent printf("Set GSM timeout timer to %d secs.\r\n",GSM_DEMO_GSM_TIMEOUT_S if (TIMER_setDuration(GSM_DEMO_GSM_TIMEOUT_TIMER_NUM,</pre>	SECS)

Figure 12-25: DemoAppGSM - Setting the timer

Document Number 1135-4713 Rev G

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

Page 88

Information classified Confidential - Do not copy (See last page for obligations)



CONFIDENTIAL

When the timer expires, the foundation code generates a TIMER event which triggers the application's processEvent() function. APL.c then executes the case TIMER in the EVENT switch. There it checks the msgSentViaGSM flag to see if a message has been sent via the terrestrial network (GPRS). If not, it reads the DIO's and sends a message via the GSM/GPRS network. It then resets the timer to expire after GSM_DEMO_GSM_TIMEOUT_SECS. After the timer expires, if a message has been sent via GSM/GPRS, the application attempts to send the DIO message via the satellite network.



Figure 12-26: DemoAppGSM - Processing a timeout

Information classified Confidential - Do not copy (See last page for obligations)

Document Number 1135-4713 Rev G

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

12.4.2 DemoAppSERIAL

This sample application demonstrates some of the Q4000/QPRO's serial port and multitasking functionality. DemoAppSerial may be used with either the AUX port, or the MTS or Logger port. It is necessary to use a different method with the AUX port than with the MTS or Logger port. DemoAppSerial first shows the use of the AUX port to send data. Note that this sample application uses network-specific calls.



ONFIDENTIAL

The AUX port

- maximum speed is 57600 bps.
- may **not** be available on certain configurations of the Q4000/QPRO. The Q4000/QPRO with Iridium or Inmarsat, for example, does not have the AUX port available.
- -1. Select the DemoAppSERIAL Workspace from the drop-down list at the top, left-hand corner of the IAR IDE screen. Open the APL.c file, as shown below



Figure 12-27: DemoAppSERIAL - Selecting the Workspace

Document Number 1135-4713 Rev G

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

Page 91

- 2. Now build, load and execute DemoAppSERIAL. The instructions for building, loading and executing the code are the same as in <u>Section 12</u>, except that after building the application, the executable bin file is: .../DemoAppSERIAL/exe/xxx-DemoAppSERIAL.bin.
- 3. After startup, check the Logger output for the line **APL DEMO:** Serial Port. This indicates that the correct DemoApp is running.

🗉 Logger - HyperTerminal
File Edit View Call Transfer Help
다 🖆 🎯 🖉 💷 🎦 🖆
No ETS_CHANNEL_MODES; default to elapsed times inactive. No SMH_WRITE_INTERVAL; default to 720 seconds. QSM_registerRxUser: SPV registered port 2 prio 85 id 3 QSM_registerRxUser: SPV registered port 1 prio 85 id 3
Starting customer application
Quake Custom Demo APL Version Aug 25 2011, 17:48:19 accessPointName = set.this.APN serverName = serverPort = 25 smtpUser =
smtprass - smtpToAddress = smtpDefaultSubject = SMTP: Invalid configuration: invalid or missing common configuration: serverName
Configure_Terr returned ERROR ERROR 0x2008:APL_init:905 ***Error starting modem NVM_vdRestoreMsnFromNVM: LL empty Enter newly created serialRxTask, with a local buffer of size 120 serialRxTask created QSM_registerRxUser: USER_APL registered port 2 prio -95 id 4 APL: Rcvd POWER ON Event
APL: Examples enabled for:
OSM_registerRxUser: TERR registered port 3 prio -100 id 1 Tx[TERR]at APL: Rcvd MTS_DTR 1 Event Telit_onOff: Turning on TERR modem

Figure 12-28: DemoAppSERIAL - Logger output that DemoAppSERIAL is running

4. It is necessary to open a Terminal Emulation program window to access the AUX port for this example. Ensure that the port is set to:

Baud rate:	115200 bps
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None

12.4.2.1 Receiving data from the AUX port

DemoAppSERIAL spawns a task to check for data on the AUX port and echoes the data back to the AUX port as it is printed. As soon as ten characters or a carriage return have been received,

Document Number 1135-4713 Rev G

CONFIDENTIAL

the data are ready to be processed. When the Serial Port example begins, APL.c waits for data to be input to the AUX port. As it is typed in, it is echoed back to the port. The application waits for ten seconds, ten characters to be input, or a carriage return to be entered. Once any of those conditions occur, the modem sends the string via either satellite or GSM/GPRS to the configured address.

The string "Testing the AUX port" was entered below.

Figure 12-29: DemoAppSERIAL - AUX output of user message

THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

Page 93

12.4.2.2 Receiving data from the MTS or Logger port

DemoAppSERIAL demonstrates how to receive data from the AUX port. In order to also receive data from either the Logger or MTS port, it is necessary to set the serial port mode in the application. There are two options for port mode: PORT_MODE_PROTOCOL and PORT_MODE_APL.

- PORT_MODE_APL passes all serial port data to the application.
- PORT_MODE_PROTOCOL gathers bytes together in packets, passes them to all foundation tasks and then to the application. This mode is preferred to allow OSI packet processing and special character trapping.

The following describes how the application is set up to receive data from the MTS port:

1. In Figure 12-30, the MTS port mode is set to PORT_MODE_PROTOCOL on line 916.

```
APL.C
    899
                return;
    900
            }
    901
    902
            //Initialize modem for SMTF
            if (initSMTPModem() == ERROR)
    903
    904
            {
    905
               LOG_logErrorMsg(__FUNCTION_, __LINE_, FAULT_QCP_MODEM_START_ERROR);
    906
            1
    907
    908
            //Initialize some serial facilites
    909
            initSer();
    910
            // Generate POWER ON Event
    911
            QEV sendEvent (POWER ON, 0, OK);
    912
    913
    914
            // Enable the checkSerRxData() function in this file to be called back when serial port
    915
            // data is received on the MTS Port
            SERIAL_setPortMode (UARTPORT_MTS, PORT_MODE_PROTOCOL, 0, 0, 0);
    916
     917
    918
            // Enable the checkSerRxData() function in this file to be called back when serial port
    919
            // data is received on the Logger Port
    920
            //APL_setSerPortMode(UARTPORT_LOGGER, PORT_MODE_PROTOCOL, 0, 0, 0);
    921
    922
            // Enter the main task loop
    923
            while(1)
    924
            {
                if (SYS_msgQReceive(aplFromQosMsgQId, (u8*)&qMsg, sizeof(qMsg), TICKS_PER_SEC) == OK)
    925
    926
                {
    927
                    if (qMsg.type != APL EVENT MSG && QLM getLogDownlink())
    928
                    ł
                        //printf("%s: Msg %d rcvd\r\n", __FUNCTION_, qMsg.type);
    929
    930
    931
                    switch (qMsg.type)
```

Figure 12-30: DemoAppSERIAL - Setting serial port mode

Page 94

2. Figure 12-31 shows that in the main processing loop, a callback function, checkSerRxData(), is called when serial port data are received (case APL_SERRXDATA_MSG).

936 // Release the memory if it has been allocated	
937 if (qMsg.eventOption.msg != NULL)	
938 {	
<pre>939 free(qMsg.eventOption.msg);</pre>	
940 }	
941	
942 break;	
943	
944 case APL_MENU_MSG:	
<pre>945 displayUtilityMenuOptions(&qMsg.menuOption);</pre>	
946 break;	
947	
948 case APL_UTIL_MSG:	
949 processUtilModeCmd(&gMsg.utilOption);	
950 break;	
951	
952 case APL_SERRXDATA_MSG:	
<pre>953 checkSerRxData(sqMsg.serRxDataOption);</pre>	
954 break;	
955	
956 case APL_FILE_MSG:	
<pre>957 verifyAndSaveFile(&qMsg.fileOption);</pre>	
958 break;	

Figure 12-31: DemoAppSERIAL - Setting serial data callback function

Document Number 1135-4713 Rev G THIS DOCUMENT CONTAINS CONFIDENTIAL AND PROPRIETARY INFORMATION OF QUAKE GLOBAL CORPORATION. IT MAY BE USED BY RECIPIENT ONLY FOR THE PURPOSE FOR WHICH IT WAS TRANSMITTED AND WILL BE RETURNED UPON REQUEST OR WHEN NO LONGER NEEDED BY RECIPIENT. DISCLOSURE TO UNAUTHORIZED THIRD PARTIES OR DUPLICATION WITHOUT THE EXPRESS WRITTEN PERMISSION OF QUAKE GLOBAL IS PROHIBITED.

CONFIDENTIAL