D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

D-Flow Firmware API
User's Guide

Emma Persson*

2013-06-05

Contents

* Phone: +46 920 52 83 19
emma.persson@d-flow.com
www.d-flow.com

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

1 Introduction

This user's guide is intended to be used together with the D-Flow Firmware API version 1.1.20130530.

1.1 Programming notes

In the UFO2 ASIC a `double` is defined as 64-bit.

In all software that should use the D-Flow Firmware API the row in listing 1 has to be present. The version number has to be the same as the kernel API used.

```
1 #pragma rtmodel = "dflow_version", "1.1.20130530"
```

Listing 1: Pragma for version control.

1.2 References

[1] D-Flow Technology AB. D-Flow UFO2 Specification - Ultrasonic Flowmeter ASIC - Customer specification. Specification, jan 2013. (Revision 0.13.2).

2 Settable parameters in ASIC

The settable parameters in the ASIC does not have any values before they are set for the first time. The default values are loaded to one or all parameters with the `ufoReloadConfigFromCode(uint8_t param)`-function. Parameter values can also be set with the `ufoSetParam(uint8_t param, uint16_t value)`-function.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

Name	Default value	Description
UL_LOOPC	20	The number of sing around loops to use in one ultrasonic measurement.
UL_SMARK	450 (28.125μs)	The beginning of the time frame when the signal should appear, expressed in 16MHz clock cycles from excitation. No signal is expected during this time.
UL_SSPACE	1000 (62.5μs)	The end of the time frame when the signal should appear, expressed in 16MHz clock cycles from SMARK. If the signal has not been detected during this time, a time-out is issued.
AG_PTRIGLEV	15	AG_PTRIGLEV is defining the threshold value for the signal. When the signal amplitude exceeds AG_PTRIGLEV the time measurement functions is triggered. AG_PTRIGLEV should be set higher than the signal noise.
AG_PTRIGLEV_LOW	31	AG_PTRIGLEV_LOW defines the threshold value used by the AGC. It should be larger than AG_PTRIGLEV.
AG_PTRIGLEV_HIGH	31	AG_PTRIGLEV_HIGH defines the last threshold value, it should be equal to or higher than AG_PTRIGLEV_LOW.
AG_STAGE1_AUTO_R1	10	Defines the gain in stage one together with AG_STAGE1_AUTO_GM1. It can be set manually but if an AGC is issued the value might change.
AG_STAGE1_AUTO_GM1	11	Defines the gain in stage one together with AG_STAGE1_AUTO_R1. It can be set manually, but if AG_STAGE1_AUTO_R1 reaches its maximum or minimum value during an AGC calibration the AG_STAGE1_AUTO_GM1 will change.
AG_STAGE2_A2	4	Defines the gain in stage two together with AG_STAGE2_R2. It will stay fixed through an AGC calibration.
AG_STAGE2_R2	4	Defines the gain in stage two together with AG_STAGE2_A2. It will stay fixed through an AGC calibration.
AG_OTHER_SEL_PAD_BP	0	Toggles whether or not the signal is output on the BP-pins. If set to 4 the signal is output, if set to 0 it is not output.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

3 API Functions

Function	Short description
ufoStart	Make a measurement loop, either use UL or SYSTEMSTATE.
useVirtTXRX	Set the tx/rx-registers to use.
ufoSaveRegs_hfc	Save the hfc data to raw data format.
ufoSaveRegs_meas	Save the measurement data to raw data format.
ufoGetHfConst	Get the value of HfConst.
ufoGetTime_std	Get the transit time in standard direction.
ufoGetTime_rev	Get the transit time in reverse direction.
ufoGetAGStatus	Get the AGC status for the measurement.
ufoTestPTRIG	Tests the flow meter by sweeping PTRIG_LEV.
ufoTestAGC	Tests the flow meter by sweeping AGC r1.
ufoGetRawData	Get the pointer to the raw data for the measurement.
ufoFetchUfoData	Convert raw data to ufo data format.
ufoGetUfoData	Get the pointer to the ufo data for the measurement.
ufoReloadConfigFromCode	Reset the software and load default values.
ufoGetParam	Get the value of a specific parameter.
ufoSetParam	Set the value of a specific parameter.
ufoLoadFPU	Load a value to the FPU.
ufoCalcFPU	Perform a calculation with the FPU.
ufoFpuCalcTransitTime	Calculate the transit times.
ufoSleep	Puts the asic to sleep.
ufoInit	Initiates the handlers.

3.1 ufoStart

```
extern uint8_t ufoStart(uint16_t setting)
```

This function sets up the software and does one measurement. Use 16-bit settings to set measurement type and directions.

In standard direction the excitation is done on TX0/TX1 (depending on measurement mode) and the signal is received on TX2/TX3. And vice versa in reverse direction. For example, when doing measurements in standard direction and measurement mode 3 the excitation is done on channels TX0 and TX1 in an alternating pattern. The signal is received on TX2/TX3 accordingly. When doing measurements in reverse direction and measurement mode 1 channel TX3 is transmitting and TX0 is receiving.

There are four measurement modes available. If more than one mode is set the lowest one will be used, e.g. if measurement modes 1 and 3 is set, measurement mode 1 will be used. For more information about the measurement modes, see [1].

By doing an AGC the automatic gain is calibrated so that the received signal has a desired signal amplitude. Doing an HFC calibrates the high frequency clock relative to the reference clock. AGC, HFC, ultrasonic measurements and ADC measurements can be used in all possible combination, but AGC and HFC should normally be done before each ultrasonic measurement.

SYSTEMSTATE is preferably used. The measurement cycles will then be controlled by hardware rather than by software. This can be especially beneficial because cpu core can be at sleep during measurement which keeps the current consumption at a minimum.

If RMIN feature is used the AGC will start at the lowest R1. If it is not set then the AGC will start at the current R1.

Gm1 can be reset to 11 at each AGC if the corresponding bit is set.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

Arguments		
uint16_t setting	16 bits with settings. Use a combination of the following values.	
	0x0004	Measure in standard direction
	0x0008	Measure in reverse direction
	0x000C	Measure in both directions
	0x0010	Do AGC
	0x0020	Do HFC
	0x0040	Do ultrasonic measurement
	0x0080	Do ADC measurement
	0x0100	Use SYSTEMSTATE
	0x0200	Set RMIN before an AGC
	0x0400	Set Gm1 to 11 before an AGC
	0x1000	Use measurement mode 1
	0x2000	Use measurement mode 2
	0x4000	Use measurement mode 3
	0x8000	Use measurement mode 4
Return		
uint8_t	Measurement status	
	UFO_OK	Measurement done ok
	UFO_AGC_FAIL	AGC failed
	UFO_HFC_FAIL	HFC failed
	UFO_MEAS_FAIL	Ultrasonic measurement failed
Prerequisites		
None		

3.2 useVirtTXRX

```
extern void useVirtTXRX(uint16_t setting)
```

Chooses the set of virtual TX/RX-registers settings to use. Virtual registers are used to store the different TX/RX settings for the different measurement modes. When a measurement mode and direction is chosen the hardware registers is set to the corresponding virtual registers. The measurement mode and direction has to be the same as when a measurement is done.

Use bits 12-15 to set measurement mode and bits 0-3 to set measurement direction(s). The other bits will be disregarded by the function.

Arguments		
uint16_t setting	16 bits with settings. Use a combination of the following values.	
	0x0004	Measure in standard direction
	0x0008	Measure in reverse direction
	0x000C	Measure in both directions
	0x1000	Use measurement mode 1
	0x2000	Use measurement mode 2
	0x4000	Use measurement mode 3
	0x8000	Use measurement mode 4
Return		
Nothing		
Prerequisites		
None		

3.3 ufoSaveRegs_hfc

```
extern void ufoSaveRegs_hfc(void)
```

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

This function calculates the hfconst and saves the hf-calibration data from the latest hfc measurement to the raw data buffer.

Arguments	
None	
Return	
Nothing	
Prerequisites	
A measurement including hfc.	

3.4 ufoSaveRegs_meas

```
extern void ufoSaveRegs_meas(uint8_t direction)
```

This function saves the measured data to the raw data buffer. The direction has to be the same as used during the measurement.

Arguments		
uint8_t direction	The direction the measurement is done in.	
	0x04	Measure in standard direction.
	0x08	Measure in reverse direction.
	0x0C	Measure in both directions.
Return		
Nothing		
Prerequisites		
An ultrasonic measurement.		

3.5 ufoGetHfConst

```
extern double ufoGetHfConst(void)
```

This function is used to read the value of the hfconst after a measurement. The hfconst is the number of internal clocks per reference clock.

Arguments	
None	
Return	
double	The value of hfconst.
Prerequisites	
ufoSaveRegs_hfc(void)	

3.6 ufoGetTime_std

```
extern double ufoGetTime_std(void)
```

This function is used to read the transit time in standard direction after the latest measurement. The transit times has to be calculated with the function `ufoFpuCalcTransitTime(void)` before it is read with this function.

Arguments	
None	
Return	
double	The transit time in standard direction.
Prerequisites	
ufoFpuCalcTransitTime(void)	

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

3.7 ufoGetTime_rev

```
extern double ufoGetTime_rev(void)
```

This function is used to read the transit time in reverse direction after the latest measurement. The transit times has to be calculated with the function `ufoFpuCalcTransitTime(void)` before it is read with this function.

Arguments	
None	
Return	
double	The transit time in reverse direction.
Prerequisites	
ufoFpuCalcTransitTime(void)	

3.8 ufoGetAGStatus

```
extern uint16_t ufoGetAGStatus(void)
```

This function is used to read the AGC status.

Arguments		
None		
Return		
uint16_t	The AGC status. The status is a combination of one or more of the following bits. Bits 10-15 are disregarded by the function.	
	Value	Description
	0x0001	Calibration finished.
	0x0002	PTRIG min trigger not reached (calibration).
	0x0004	PTRIG min trigger not reached (direction 0).
	0x0008	PTRIG min trigger not reached (direction 1).
	0x0010	PTRIG max trigger reached (calibration).
	0x0020	PTRIG max trigger reached (direction 0).
	0x0040	PTRIG max trigger reached (direction 1).
	0x0080	PTRIG is ok (calibration).
	0x0100	PTRIG is ok (direction 0).
	0x0200	PTRIG is ok (direction 1).
Prerequisites		
None		

3.9 ufoTestPTRIG

```
extern void ufoTestPTRIG(uint16_t setting)
```

This functions sweeps PRIG_LEV. For each PTRIG_LEV a measurement is done and the transit times are printed with printf. Whether or not the measurement is successful, the AGC status is printed. The AGC status is described in section 3.8

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

Arguments		
uint16_t setting	16 bits settings. Use a combination of the following values. Bits 4-11 are disregarded by the function.	
	0x0004 0x0008 0x000C	Measure in standard direction. Measure in reverse direction. Measure in both directons.
	0x1000 0x2000 0x4000 0x8000	Use measurement mode 1. Use measurement mode 2. Use measurement mode 3. Use measurement mode 4.
Return		
Nothing		
Prerequisites		
None		

3.10 ufoTestAGC

```
extern void ufoTestAGC(uint16_t setting)
```

This function sweeps the AGC r1-value. For each step in r1 a measurement is done. The transit times from these measurements are printed with printf. Whether or not the measurement is successful, the AGC status is printed. The AGC status is described in section 3.8.

Arguments		
uint16_t setting	16 bit settings. Use a combination of the following values. Bits 4-11 are disregarded by the function.	
	0x0004 0x0008 0x000C	Measure in standard direction. Measure in reverse direction. Measure in both directions.
	0x1000 0x2000 0x4000 0x8000	Use measurement mode 1. Use measurement mode 2. Use measurement mode 3. Use measurement mode 4.
Return		
Nothing		
Prerequisites		
None		

3.11 ufoGetRawData

```
extern uint8_t * ufoGetRawData(uint8_t seqnr)
```

This function is used to get the pointer to the raw data to read. The sequence number can be used by PC software to keep track of the measurements.

Arguments	
uint8_t seqnr	Sequence number.
Return	
uint8_t *	Pointer to raw data.
Prerequisites	
None	

3.12 ufoFetchUfoData

```
extern void ufoFetchUfoData(void)
```

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

This function is used to convert raw data to ufo data.

Arguments	
None	
Return	
Nothing	
Prerequisites	
ufoFpuCalcTransitTime(void)	

3.13 ufoGetUfoData

```
extern uint8_t * ufoGetUfoData(uint8_t seqnr)
```

This function is used to get the pointer to the ufo data to read. The sequence number can be used to keep track of the measurements.

Arguments	
uint8_t seqnr	Sequence number.
Return	
uint8_t *	Pointer to ufo data.
Prerequisites	
ufoFetchUfoData(void)	

3.14 ufoReloadConfigFromCode

```
extern void ufoReloadConfigFromCode(uint8_t param)
```

This functions loads the default values for one or all parameters. There is a description of the parameters and their default values in section 2.

Arguments	
uint8_t param	0 for resetting all parameters to their default values. Use the parameter names from section 2 to reset the corresponding parameter.
Return	
Nothing	
Prerequisites	
None	

3.15 ufoGetParam

```
extern uint16_t ufoGetParam(uint8_t param)
```

This function gets the value of the wanted parameter. There is a description of the parameters in section 2.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

Arguments	
uint8_t param	The name of the parameter to get the value of.
	UL_LOOPC UL_SMARK UL_SSPACE AG_PTRIGLEV AG_PTRIGLEV_LOW AG_PTRIGLEV_HIGH AG_STAGE1_AUTO_R1 AG_STAGE1_AUTO_GM1 AG_STAGE2_A2 AG_STAGE2_R2 AG_OTHER_SEL_PAD_BP
Return	
uint16_t	The value of the input parameter.
Prerequisites	
None	

3.16 ufoSetParam

```
extern void ufoSetParam(uint8_t param, uint16_t value)
```

This function sets the value to the specified parameter. There is a description of the parameters in section 2.

Arguments		
uint8_t param uint16_t value	The name of the parameter to set the value on. The value to set to the parameter.	
	Param	Valid values
	UL_LOOPC	0-65535
	UL_SMARK	0-65535
	UL_SSPACE	0-65535
	AG_PTRIGLEV	0-31
	AG_PTRIGLEV_LOW	0-31
	AG_PTRIGLEV_HIGH	0-31
	AG_STAGE1_AUTO_R1	0-31
	AG_STAGE1_AUTO_GM1	0-15
	AG_STAGE2_A2	0-7
	AG_STAGE2_R2	0-31
	AG_OTHER_SEL_PAD_BP	4: Signal output on bp<1:0> 0: No signal output on bp<1:0>
Return		
Nothing		
Prerequisites		
None		

3.17 ufoLoadFPU

```
extern void ufoLoadFPU(double a)
```

Arguments	
double a	The operand to load to the FPU.
Return	
Nothing	
Prerequisites	
None	

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

3.18 ufoCalcFPU

```
extern double ufoCalcFPU(double a, double b, fpu_op_t op)
```

This function is used to do calculations with the fpu.

Arguments	
double a double b fpu_op_t op	The first operand for the calculation. The second operand for the calculation. The operation to perform on the operands. The possible operations are: FPU_OP_ADD FPU_OP_SUB FPU_OP_MUL FPU_OP_DIV
Return	
double	The result of the calculation.
Prerequisites	
None	

3.19 ufoFpuCalcTransitTime

```
extern void ufoFpuCalcTransitTime(void)
```

This function is used to calculate the transit times. This has to be run after an ultrasonic measurement before the result is converted to the output format.

Arguments	
None	
Return	
Nothing	
Prerequisites	
ufoSaveRegs_hfc(void) ufoSaveRegs_meas(uint8_t direction) The hfc and ultrasonic measurement does not have to be done at the same instant, but at least one of each should be made before the ufoFpuCalcTransitTime(void) function is called.	

3.20 ufoSleep

```
extern void ufoSleep(void)
```

This function puts the asic to sleep. To use the sleep function SYSTEMSTATE has to be used when doing measurement. Also the wakeup triggers has to be set.

Arguments	
None	
Return	
Nothing	
Prerequisites	
None	

3.21 ufoInit

```
extern void ufoInit(void)
```

This function initiates the asic.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

Arguments	
None	
Return	
Nothing	
Prerequisites	
sys_state_module_enable(sys_state_module_t modules, bool enable) for each of the four modules SYS_STATE_MODULE_AD_CTRL SYS_STATE_MODULE_UL_CTRL SYS_STATE_MODULE_AG_CTRL SYS_STATE_MODULE_TX_CTRL	

4 Programming examples

4.1 Programming example 1: Transit time measurement

This first programming example shows how to use the D-Flow Firmware API to do an ultrasonic measurement in measurement mode 1. The main function is shown in listing 2. The function `initSerialModule()`, listing 4, is needed to initiate the serial communication. The function `putchar()`, listing 6, is used by the function `printf()`.

```
#include <stdio.h>
#include "api.h"
#include "ufo-common.h"

//Has to be same version number as kernel API for software to compile.
#pragma rtmodel = "dflow_version", "1.1.20130530"


void main(void)
{
  // local variables
  static uint8_t status;
  register uint16_t i;

  // Initialize vic
  vic_handler_init();

  // Initialize sys_state
  sys_state_handler_init();

  // enable used modules
  sys_state_module_enable(SYS_STATE_MODULE_AG_CTRL, true);
  sys_state_module_enable(SYS_STATE_MODULE_UL_CTRL, true);
  sys_state_module_enable(SYS_STATE_MODULE_TX_CTRL, true);

  ufoInit();         // initialize library
  ufoReloadConfigFromCode(0);    // load default values
  ufoSetParam(UL_LOOPC,           1);  // set customized value
  ufoSetParam(UL_SMARK,         320);  // set customized value
  ufoSetParam(UL_SSPACE,        480);  // set customized value
  ufoSetParam(AG_STAGE1_AUTO_R1, 3);  // set customized value

  // for serial output
  initSerialModule();
```

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

```
35
36    // Make measurements
37    while(1) {
38
39      status = ufoStart(BIT_MEAS_CFG_0 +
40            BIT_MEAS_ATR_AGC_RMIN +
41            BIT_MEAS_ATR_SYSSTATE +
42            BIT_MEAS_ACT_SING +
43            BIT_MEAS_ACT_HFC +
44            BIT_MEAS_ACT_AGC);
45
46      printf("AGC: (DIR_STD) (AG status=%hu) ",ufoGetAGStatus());
47      printf("(r1=%hu, gm1=%hu) ",
48             ufoGetParam(AG_STAGE1_AUTO_R1),
49             ufoGetParam(AG_STAGE1_AUTO_GM1));
50
51      printf("HFC: hfConst = %4.0f ",ufoGetHfConst());
52
53      ufoCalcTransitTime();
54      printf("SING: std=%0.9f rev=%0.9f us (AG status=%hu)",1e6*ufoGetTime_std(),
55              1e6*ufoGetTime_rev(),ufoGetAGStatus());
56      printf("\n");
57
58      i = 32000;
59      while(--i); // 10ms?
60    }
61
62  }
```

Listing 2: Transit time measurement

4.2 Programming example 2: Transit time measurement with power down

This programming example is for demonstrating how to do an ultrasonic measurement with power down implemented. The main function is seen in listing 3. The function `initSerialModule()`, listing 4, is used to initiate the serial communication. The function `setup_sysstate()`, listing 5, is used to set up systemstate for power down mode. Also, the `putchar()`-function in listing 6 has to be used for the `printf()`-function to work properly.

```
1  #include <stdio.h>
2  #include "api.h"
3  #include "ufo-common.h"
4
5  //Has to be same version number as kernel API for software to compile.
6  #pragma rtmodel = "dflow_version", "1.1.20130530"
7
8  void main(void)
9  {
10   // local variables
11   static sys_state_core_wakeup_status_t  core_wakeup_status;
12   static sys_state_meas_wakeup_status_t  meas_wakeup_status;
13
14   // get wakeup reason
15   meas_wakeup_status = sys_state_get_meas_wakeup_status();
```

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

```
16  core_wakeup_status = sys_state_get_core_wakeup_status();
17
18  // if rtc is triggered, means that the ASIC has been configured
19  if (meas_wakeup_status.bf.rtc_trigger == 1) {
20
21    initSerialModule();
22
23    // A measurement is finished, get values and process them ...
24    ufoSaveRegs_hfc();
25    printf("hfConst=%f,\t",ufoGetHfConst());
26    printf("AGC (r1=%hu, gm1=%hu)\t",
27           ufoGetParam(AG_STAGE1_AUTO_R1),
28           ufoGetParam(AG_STAGE1_AUTO_GM1));
29    ufoSaveRegs_meas(BIT_DIR_BOTH);
30    ufoFpuCalcTransitTime();
31
32    printf("up=%f\tdown=%f us (AG status=%hu)\n",1e6*ufoGetTime_std(),
33      1e6*ufoGetTime_rev(),ufoGetAGStatus());
34
35    while (uart_transmitting(UART0)); //busy wait until printf is finished
36
37    ufoSleep();
38  }
39
40  //
41  // Only gets here if powered up first time
42  //
43
44  // Initialize vic
45  vic_handler_init();
46
47  // Initialize sys_state
48  sys_state_handler_init();
49
50  // Make sure its powered up first time
51  if (core_wakeup_status.bf.power_up == 1){
52    sys_state_module_enable(SYS_STATE_MODULE_AG_CTRL, true);
53    sys_state_module_enable(SYS_STATE_MODULE_UL_CTRL, true);
54    sys_state_module_enable(SYS_STATE_MODULE_TX_CTRL, true);
55
56    ufoInit();        // initialize library
57    ufoReloadConfigFromCode(0);   // load default values
58    ufoSetParam(UL_LOOPC,          1);  // set customized value
59    ufoSetParam(UL_SMARK,        320);  // set customized value
60    ufoSetParam(UL_SSPACE,       480);  // set customized value
61    ufoSetParam(AG_STAGE1_AUTO_R1, 3);  // set customized value
62
63    setup_sysstate();     // setup systemstate
64  }
65
66  // Should never get here
67  while(1);
68
```

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

```
69 } // end main()
```

Listing 3: Transit time measurement with power down.

4.3 Functions used by the main-functions

```
1  void initSerialModule(void) {
2
3    // Enable modules
4    sys_state_module_enable(SYS_STATE_MODULE_UART0, true);
5    sys_state_module_enable(SYS_STATE_MODULE_IOMUX_CTRL, true);
6
7    // Select UART0 function instead of GPIO 0-1
8    iomux_ctrl_select((iomux_io_t)(IOMUX_IO_0 + IOMUX_IO_1), IOMUX_FUNC_2ND);
9
10   uart_handler_init(UART0);
11   uart_clear_all_irq_events(UART0);  // clear pending irqs
12
13
14   uart_init(UART0, UART_SPEED_57600, UART_FRAC_57600, UART_DATA_LEN_8,
15                    UART_PARITY_NONE, UART_ONE_STOP);
16   uart_set_rx_enable(UART0, true);
17   uart_loopback_enable(UART0, false);
18 }
```

Listing 4: Function to initiate serial communication.

```
1  void setup_sysstate(void)
2  {
3    printf("Enable infinite loop of US meas (MCU off during meas)\n");
4
5    // config wakeups
6    sys_state_config_meas_wakeup(SYS_STATE_WAKEUP_BY_RTC_TRIG);
7    sys_state_config_core_wakeup(SYS_STATE_WAKEUP_BY_MEAS_FIN);
8
9    // configure clock wait
10   //sys_state_config_hclk_delay(30, 30);
11
12   // configure measurement (refclk_cal, agc_cal, adc_meas, us_meas)
13   sys_state_config_meas_flow(true, true, false, true);
14
15   // measure in both directions
16   ufoSetBothDir(true);
17
18   // set mode
19   useVirtTXRX(BIT_MEAS_CFG_0 + BIT_MEAS_DIR_STD); // use mode1
20
21   ufoSleep();
22 }
```

Listing 5: Function to initiate systemstate.

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB

D-Flow Technology AB
Gammelstadsvägen 5 D
SE - 972 41 LULEÅ
SWEDEN
+46 920 52 83 00

info@d-flow.com
www.d-flow.com

```
1 int putchar(int value)
2 {
3   if (value == '\n')
4     uart_transmit_byte(UART0, '\r', true); //Convert EOL to CR/LF
5   uart_transmit_byte(UART0, (uint8_t)value, true);
6   return value;
7 }
```

Listing 6: Function used by printf().

PROPRIETARY CLAUSE: This document contains information and/or data which is proprietary and confidential to D-Flow Technology AB. This information and/or data is to be used only for the benefit of the supplier, and is not to be used for the benefit of other parties without the written consent of D-Flow Technology AB.

© 2013 D-Flow Technology AB