

hsk-ecu-stub

65

Generated by Doxygen 1.8.12

Contents

- 1 HSK XC878 Electronic Control Unit Stub 1**

- 2 File Index 3**
 - 2.1 File List 3

- 3 File Documentation 5**
 - 3.1 config.h File Reference 5
 - 3.1.1 Detailed Description 6
 - 3.1.2 Macro Definition Documentation 6
 - 3.1.2.1 CAN0_BAUD 6
 - 3.1.2.2 CAN0_IO 6
 - 3.1.2.3 CAN1_BAUD 6
 - 3.1.2.4 CAN1_IO 6
 - 3.1.2.5 CLK 6
 - 3.2 main.c File Reference 7
 - 3.2.1 Detailed Description 7
 - 3.2.2 Function Documentation 7
 - 3.2.2.1 init() 7
 - 3.2.2.2 main() 8
 - 3.2.2.3 run() 8

- Index 9**

Chapter 1

HSK XC878 Electronic Control Unit Stub

This project is a stub to clone when starting the development of a new device.

The `hsk-libs` repository is included as a submodule, a `git submodule update` may be necessary to populate it.

See the `Makefile` for documentation on build parameters and `Makefile.local` to override defaults locally.

After setting the new project up, run `uVisionupdate.sh` to update the μ Vision project file. The `uVisionupdate.sh` script also generates the list of ISR callbacks for μ Vision's call tree/overlying engine.

The list generation only recognizes direct assignments to `hsk_isr<number>.<SOURCE>` and calls with function pointer arguments to:

- `hsk_timer[01]_setup()`
- `hsk_ex_channel_enable()`

More complex assignments might require an update of `hsk-libs/scripts/overlays.awk`.

See also

[PDF Version](#)

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

config.h	Configuration for the Infineon XC800 Starter Kit	5
main.c	Device main.c Stub	7

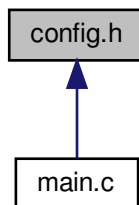
Chapter 3

File Documentation

3.1 config.h File Reference

Configuration for the Infineon XC800 Starter Kit.

This graph shows which files directly or indirectly include this file:



Macros

- #define `CLK` 8000000UL
The external oscillator clock frequency.
- #define `CAN0_BAUD` 1000000
The CAN0 baud rate in bits/s.
- #define `CAN1_BAUD` 1000000
The CAN1 baud rate in bits/s.
- #define `CAN0_IO` CAN0_IO_P10_P11
The CAN0 IO pin configuration RX P1.0 TX P1.1.
- #define `CAN1_IO` CAN1_IO_P14_P13
The CAN1 IO pin configuration RX P1.4 TX P1.3.

3.1.1 Detailed Description

Configuration for the Infineon XC800 Starter Kit.

Author

kami

3.1.2 Macro Definition Documentation

3.1.2.1 CAN0_BAUD

```
#define CAN0_BAUD 1000000
```

The CAN0 baud rate in bits/s.

3.1.2.2 CAN0_IO

```
#define CAN0_IO CAN0_IO_P10_P11
```

The CAN0 IO pin configuration RX P1.0 TX P1.1.

3.1.2.3 CAN1_BAUD

```
#define CAN1_BAUD 1000000
```

The CAN1 baud rate in bits/s.

3.1.2.4 CAN1_IO

```
#define CAN1_IO CAN1_IO_P14_P13
```

The CAN1 IO pin configuration RX P1.4 TX P1.3.

3.1.2.5 CLK

```
#define CLK 8000000UL
```

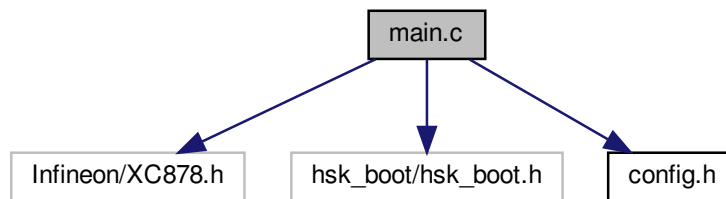
The external oscillator clock frequency.

3.2 main.c File Reference

Device [main.c](#) Stub.

```
#include <Infineon/XC878.h>
#include <hsk_boot/hsk_boot.h>
#include "config.h"
```

Include dependency graph for main.c:



Functions

- void [main](#) (void)
Call init functions and invoke the run routine.
- void [init](#) (void)
Initialize ports, timers and ISRs.
- void [run](#) (void)
The main code body.

3.2.1 Detailed Description

Device [main.c](#) Stub.

Author

kami

3.2.2 Function Documentation

3.2.2.1 init()

```
void init (  
    void )
```

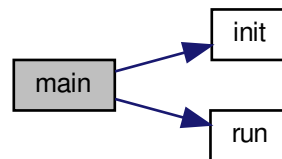
Initialize ports, timers and ISRs.

3.2.2.2 main()

```
void main (  
    void )
```

Call init functions and invoke the run routine.

Here is the call graph for this function:



3.2.2.3 run()

```
void run (  
    void )
```

The main code body.

Index

- CAN0_BAUD
 - config.h, [6](#)
- CAN0_IO
 - config.h, [6](#)
- CAN1_BAUD
 - config.h, [6](#)
- CAN1_IO
 - config.h, [6](#)
- CLK
 - config.h, [6](#)
- config.h, [5](#)
 - CAN0_BAUD, [6](#)
 - CAN0_IO, [6](#)
 - CAN1_BAUD, [6](#)
 - CAN1_IO, [6](#)
 - CLK, [6](#)
- init
 - main.c, [7](#)
- main
 - main.c, [7](#)
- main.c, [7](#)
 - init, [7](#)
 - main, [7](#)
 - run, [8](#)
- run
 - main.c, [8](#)