

MPU

**Main Processing Unit
of
Mur SAT**

Roland Sahlsten

FH JOANNEUM

Kapfenberg, Austria

`roland.sahlsten.ase10@fh-joanneum.at`

1 Introduction

MurSat is a sub-project of an art project in Graz, which tries to give its visitors some insight into the world of technique. The project tries to combine art and technique in an exhibition in Graz, but also through a very small satellite, also called nano-satellite or tube-sat, traveling around mother earth in a low orbit. MurSat is also the name of this nano-satellite, which is developed and assembled by a team of engineering students. In march 2012 the satellite will be launched and brought to an orbit in a distance to the earths surface of about 310 km. It will stay there for 4-6 weeks and travel at a speed of approximately 23.000 km/h. Then it will come closer to earth, loose speed and finally enter the earths atmosphere where it will burn up. During the time in orbit, it will hold radio contact with several base stations in Austria, Germany and Spain. It will also take pictures which will be sent to earth whenever the satellite passes by on of the base stations. The satellite is built up of several units. There is a communication unit, a house holding unit, responsible for the electric energy on board, a main processing unit and several sensors. My tasks in this project are the development and implementation of the main processing unit. It is powered by an ARM M3 micro processor running a free real time operating system. The entire software for the main processing unit is written in ANSI C and compiled and built using the GNU ARM tool chain running in the eclipse IDE.

2 Related Work

It is not possible to navigate the satellite since it has no rocket propulsion built in. Therefore it is also not possible to aim the camera to a certain object. This circumstance makes it quite hard to take pictures.

So the first consideration is just to take any pictures and send them from the satellite down to the base stations on earth. But this method is not very feasible for some reasons. First we need to take a lot of pictures to get at least some good ones. This would need a lot of memory on the satellite, which is not such a big deal, but the writing of data needs

electric energy which is quite limited on the satellite. The second problem is, that we also would need to transfer a big amount of data. Data transfer is limited through transfer rate and also through the time windows that allow communication with the satellite.

It is quite obvious that there is a need of preselection of the pictures which shall be downloaded from the satellite. So we thought about a simple algorithm which could be able to do some evaluation if a picture whether it is 'interesting' or not. Totally black and totally white pictures can be sorted out. Other pictures could show something interesting. The satellite travels on a n orbit which is about 318 km above the earths surface. Therefore the dihedral angle of earth in view is about **TOTO: check raumwinkel** which is rather big compared to the one which is showing the sun. This means, that the likelihood that a picture, which is not only black or white, is rather big that it shows some parts of the earth.

In order to raise the likelihood of shooting a picture at the right moment, we were searching for a way of how to find this right moment. Using the magnetic field of the earth is not an appropriate approach since the magnetic force lines of the earth run approximately parallel to the earth surface (except of near the poles), which would not provide enough information to calculate the satellites orientation in a rather easy way. Furthermore, a teslameter that provides values of a usable accuracy would be to expensive as well as too damageable.

Finally we decided to use a 3D - light sensor. It shall help to find out the relative orientation of the satellite to the earth. One photodiode on each side of a cube would provide 6 lines of different light intensities. When the satellite is spinning through space, the photodiodes will give periodical signals, such way that three different periodes will be seen. The satellite will rotate around three axis, whereby each of these three periods corresponds to the rotation around one of these axes.

In Figure 1 we can see, that these periodes have one common periode, which is starting at t_0 . The point im time t_p (here just an example) can the be taken as the point in time which is perfect for taking a picture. So the idea is to cumulate the photo-sensors measurements and download them to earth.

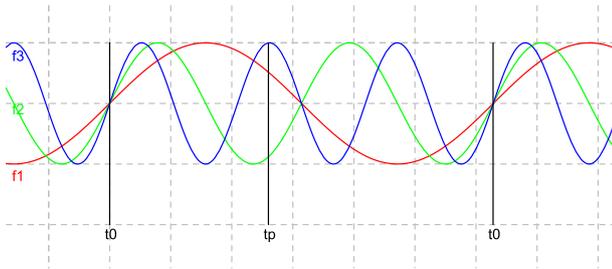


Figure 1: Periodic output of 3 photodiodes.

After analysing and calculating a certain point in time t_p , the satellite can be asked to take pictures at t_p . If these pictures are not satisfying, t_p can be tuned until the results fit to the expected outcome.

3 System Overview

The satellite system consists of several units. All of these parts are needed to fulfill the tasks the satellite shall take. The satellite shall be able to communicate via radio, is equipped with a camera an Alien-Encounter and a microphone, almost like a bluetooth webcam. Furthermore there are 3 light-sensors and 4 temperature sensors on board. Please refer to figure 2 to get an overall picture.

As shown in the Figure 2, there are many parts.

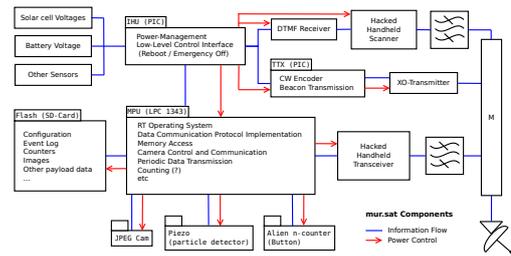


Figure 2: System Component Overview.

4 Operating System

The software running on the MPU is based on a preemptive real time operating systems and manages the tasks the satellite has to fulfill. The Operating system, on which the whole software is based on, is FreeRTOS.

<http://www.freertos.org>

4.1 FreeRTOS

FreeRTOS is a preemptive multitasking real time operating system. It provides methods for creating and scheduling tasks and implements queues and semaphores for communication between, and synchronization of the tasks.

4.2 MPU Tasks

The MPU gets commands from the low level control interface IHU and from the Hackled Handheld Transceiver. The IHU sends mainly commands which are related to the power manage-

ment, whereby the Hackeld Handheld Transceiver is the command communication interface between the satellite and a base station on earth. The satellite has to send its bacon signal permanently. This way it can be located in space. When the satellite gets in communication distance to a base station on earth, a special signal is sent to the satellite which is received by the IHU via the Hacked Handheld Scanner. The IHU then tells the MPU to swith on the Hacked Handheld Transceiver. From now on a communication between the base station and the satellite is possible.

The satellite can receive configuration commands and data request commands from the base station. The configuration commands are used to set parameters like e.g. the sampling rate of the microphone (piezzo element), or to set parameters for the camera, as well as to tell the satellite when and how often to take pictures.

4.2.1 Messages

The communication inside the operatin system is based on a message system. Tasks can send messages to each others.

FreeRTOS provides with an object called xQueue. This is used to communicate between tasks in FreeRTOS. An xQueue is primarily a void pointer which can be sent to a task. This pointer can point to any address in the memory. In this implementation xQueues are used to point to structures which represent "Messages".

```
1 typedef struct
2 {
3     void      *pData;
4     Sender_t  Sender;
5 } Message_t;
```

A message consists of a pointer to the data which shall be sent and an identifier of the sender. The Sender type is just an enumeration of the tasks that want to communicate with each others.

4.2.2 Kernel Process Task

The Kernel Process Task is responsible for dispatching the jobs and assigning them to the according tasks. Each time a communication interface (UART, SPI, I2C) receives data, its interrupt service routine notifies the Communication Process

about this event. The Communication Process collects all arriving data and when a comand is completely received, the Kernel Process Task is notified, gets the command via a message, checks this command and sends the respective message to the task which is responsible for the execution of the command.

4.2.3 IHU Process Task

It seems that the IHU Process Task is not needed, so it will be taken out of the operating system. Actually the Internal Householding Unit IHU is providing the satellite with energy which is a rather limited ressource. Therefore it is necessary to implement some kind of protocol to manage the energy consumption of all satellite components. This IHU Process Tasks was ment to handle this protocol, but right now it looks like if this task is taken over by the Kernel Processing Task.

4.2.4 Communication Process Task

This tasks purpose is to receive commands and send responses over the respecting interface (UART, SPI, I2C). When ever a byte arrives at one of the interfaces, the respective interrupt service routine ISR is triggert. The ISR copies the received byte to a buffer and if the end of frame EOF character is received, it sends a message to the communication task. The communication task then analyses the received frame and informs the Kernel Process Task. The Communication Task can also receive messages from other tasks. These messages contain data frames that shall be sent via one of the interfaces. When such a message is received by the Communication Task, it sends the messages data over the respective interface.

4.2.5 Alien Encounter Process Task

The Alien Encounter Task is waiting for a message which is sent by the interrupt service routine of the according external interrupt. When ever an alien is passing by the satellite who is willing to press the Alien Encounter Button, and also can manage to do so, an external interrupt is triggert.

4.2.6 Light Sensor Process Task

The Light Sensor Process Task is responsible for handling light sensor related jobs. The light sensors are placed such way that they detect light in all three dimmensions in relation to the satellites local coordinate system. When sampling the intensity of light for these directions it shall be possible to calculate the satellites, especially the cameras orintation related to sun and earth. This should help to decide at which moment a picture of the earth can be taken. The sampling rate of the light sensors can be adjusted and is depending of the angular velocity of the satellites rotation.

4.2.7 Camera Process Task

The Camera Process Task handels all camera related jobs. It triggers the camera to take pictures and reads the cammeras data (pictures) from the cameras memory to a buffer. When ever the buffer is filled, this task sends a message to the Memory Process Task to store the data in the SD-Card.

4.2.8 Memory Process Task

The responsibility of the Memory Process Task is to read and write data from and to the SD-Card. This task can receive messages from any other task. If a message sender wants to write data to the SD-Card, the Memory Process Task reads the data from the message and sends it to the SD-Card. If a task wants to read data from the SD-Card, the Memory Process Task reads the desired data from the SD-Card and writes it to a message which is then sent to the requesting task.

4.3 Component Architecture

The software consists of several components. Each component has its dedicated functionality as described before.

Figure 3 shall give an overview of the software components and how thea are related to each others.

4.3.1 Kernel Process Task

[Put detailed description here.](#)

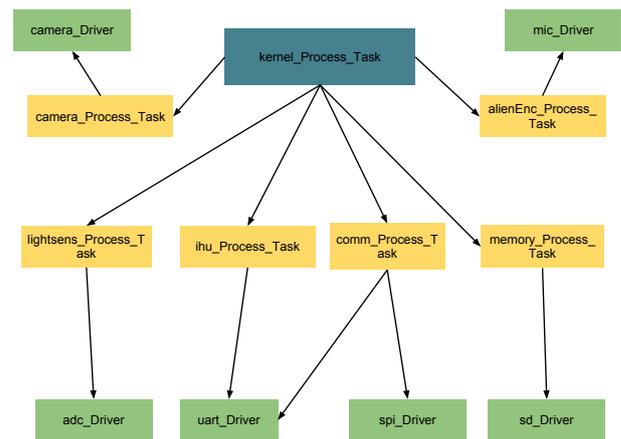


Figure 3: Software System Overview.

4.3.2 IHU Process Task

[Put detailed description here.](#)

4.3.3 Light Sensor Process Task

[Put detailed description here.](#)

4.3.4 Camera Process Task

[Put detailed description here.](#)

4.3.5 SDCard Process Task

[Put detailed description here.](#)