# HICS - Sistema de Aquisição de Dados

## Especificações Técnicas

### Versão 2.0

### RELATÓRIO FINAL

### 24-08-2020

## Sumário

# 1. Objetivo

O objetivo deste documento é descrever o sistema de aquisição de dados do telescópio HATS (High Altitude THz Solar photometers) que está sendo desenvolvido pelo Centro de Rádio-Astronomia e Astrofísica Mackenzie (CRAAM).

O HATS é um telescópio a ser instalado em solo em grande altitude para observar em uma banda de frequências que é fortemente atenuada pela absorção atmosférica. Para uma descrição do HATS ver [RD01].

O telescópio possui uma célula de Golay junto com um chopper de calibração como elemento detector. A célula de Golay gera um sinal analógico de baixa frequência (20Hz). Este sinal analógico precisa ser lido por um sistema de aquisição de dados. Este sistema estará instalado na estrutura do telescópio e será denominado como *HATS Interface & Control System* (HICS).

Neste documento estão descritos os componentes de hardware e a arquitetura do firmware deste sistema, bem como o ambiente de programação utilizado para o desenvolvimento do projeto.

## 2. Documentos de Referência

| ID | Autor | Agência | Nome do Documento | Título |
|---|---|---|---|---|
| RD01 | Guillermo Giménez de Castro | CRAMM | CRAAM-Especificacoes_Tecnicas_HATS.pdf | Descrição e Especificações Técnicas do HATS Versão 2.1 |
| RD02 | ATMEL | ATMEL | Atmel-42216-SAM4E-Xplained-Pro_User-Guide.pdf | Manual do usuário da placa ATMEL SAM4E Xplained-Pro |
| RD03 | Analogo Device | Analog Device | EVAL-AD7770FMCZ-7771FMCZ-7779FMCZ-UG-884 | Manual do usuário da placa EVAL-AD7770 |

## 3. Lista de Acrônimos e Abreviações

| | |
|---|---|
| ADC | Conversor de sinal analógico para digital |
| ASF | Atmel Software Framework |
| GCC | GNU Compiler Collection |
| HATS | High Altitude THz Solar photometers |
| HICS | HATS Interface and Control System |
| HOMS | HATS Operation, Monitoring & Storage |
| KSPS | Kilo Samples per Second |
| SNTP | Simple Network Time Protocol |
| PGA | Programmable Gain Amplifier |
| SPI | Serial Peripheral Interface |

# 4. Especificações Técnicas do Sistema de Aquisição de Dados do HICS

O sistema de aquisição de dados do HICS é composto por duas placas, sendo uma placa utilizada para fazer a amostragem do sinal analógico da célula de Golay e outra placa utilizada para fazer o controle e processamento dos sinais amostrados. Estas placas estão descritas abaixo.

## 4.1 Diagrama em Blocos

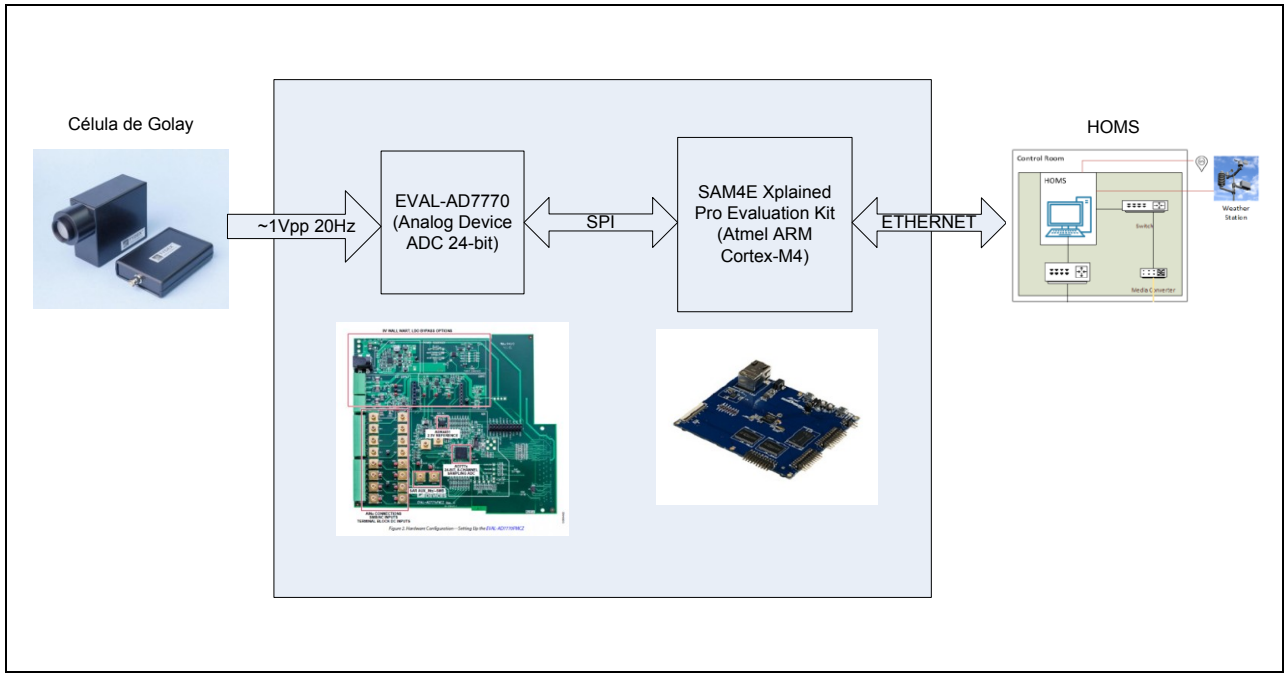



Figura: Diagrama em Blocos do Sistema de Aquisição de Dados do HICS

## 4.2 Placa microcontrolador SAM4E Xplained Pro

O processamento do sistema de aquisição de dados é constituído de uma placa da Atmel modelo **SAM4E Xplained Pro** [RD02]. Esta placa é baseada no microcontrolador ARM Cortex M4 Atmel ATSAM4E16E.

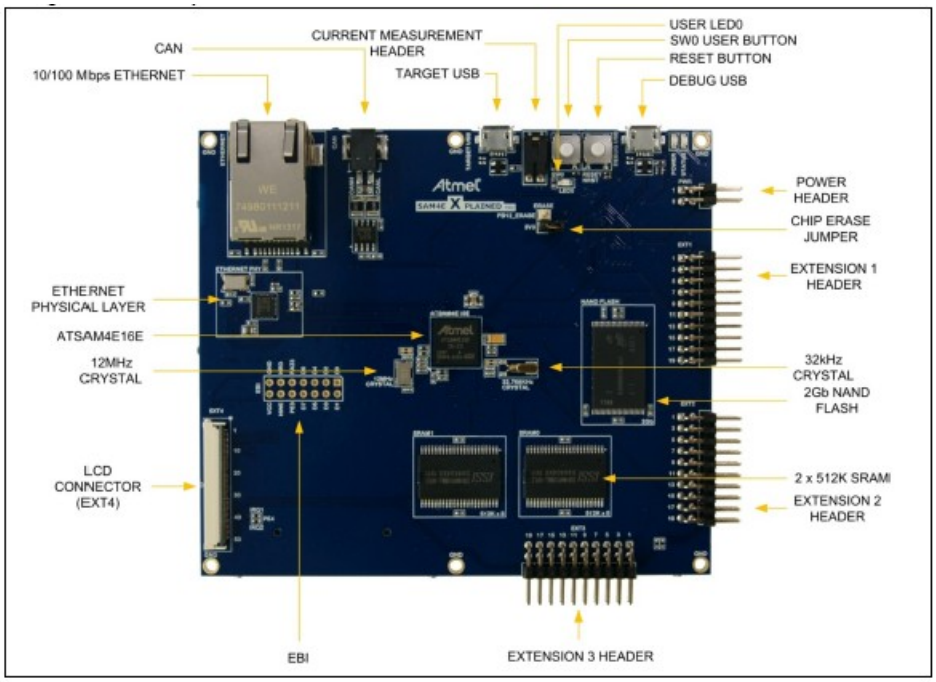



Figura: Placa de Desenvolvimento ATMEL SAM4E Xplained Pro

A principais características desta placa para este projeto são:

- Microcontrolador de alta performance 32-bit ARM Cortex M4 RISC modelo Atmel ATSAM4E16E
- Unidade de Ponto Flutuante (FPU)
- Instrução DSP
- 2 Kbytes Cache rodando até 120MHz
- Memória Flash de 1024KB
- Memória SRAM de 128KB
- Embedded Debugger (EDBG)
- Interface Ethernet 10/100-T
- Interface USB

- Conector de expansão com interface SPI
- Memória externa NAND Flash de 2Gb 8-bit
- Memória externa SRAM Dual 512K 8-bit
- RTC com calendários Gregorian e Persian

## 4.3 Placa de Aquisição de Dados EVAL-AD7770 ADC 24-bit

A leitura do sinal analógico da Célula de Golay é realizada pela placa da Analog Device modelo EVAL-AD7770 [RD03] que possui 8 canais para conversão simultânea de sinal analógico para digital com resolução de 24-bit.

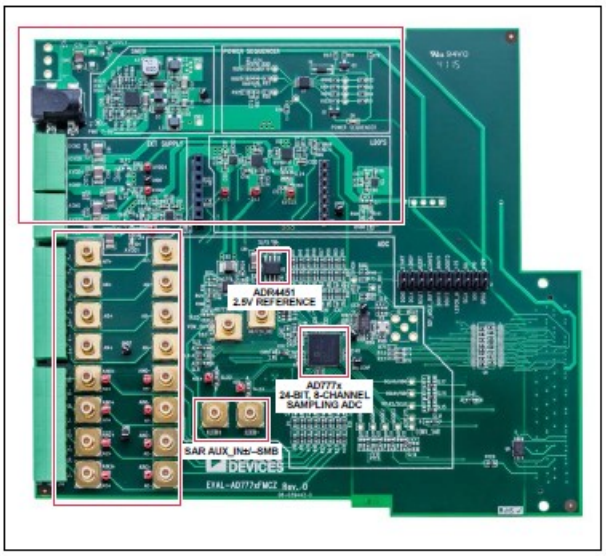



Figura: Placa Analog Device EVAL-AD7770 ADC 24-bit

As principais características desta placa para este projeto são:

- Conversor de sinal analógico para digital com resolução de 24-bit
- Conversão simultânea de 8 canais
- Amplificador de ganho programável de 1x, 2x, 4 e 8x (PGA)
- Tensão de referência de 2.5V de alta precisão e baixo ruído
- Taxa de amostragem de 32 mil amostras por segundo (KSPS)
- Range dinâmico de 103dB
- Distorção Harmônica total de -109 dB
- Coeficiente de temperatura de ±10 ppm/°C
- Interface de comunicação SPI

Como esta placa possui 8 canais, também será possível expandir o sistema para realizar outras leituras de sinais além da célula de Golay.  Além da Célula de Golay, a tensão de alimentação do sistema está sendo monitorada por um dos canais desta placa.

## 5.0 Ambiente de Desenvolvimento do Firmware

Para o desenvolvimento do firmware foi utilizado a plataforma gratuita de desenvolvimento Atmel Studio. Esta plataforma oferece um ambiente de desenvolvimento integrado com editor de código, compilador, debug, programador e uma vasta biblioteca de códigos exemplos que podem ser utilizados como referência para o projeto.

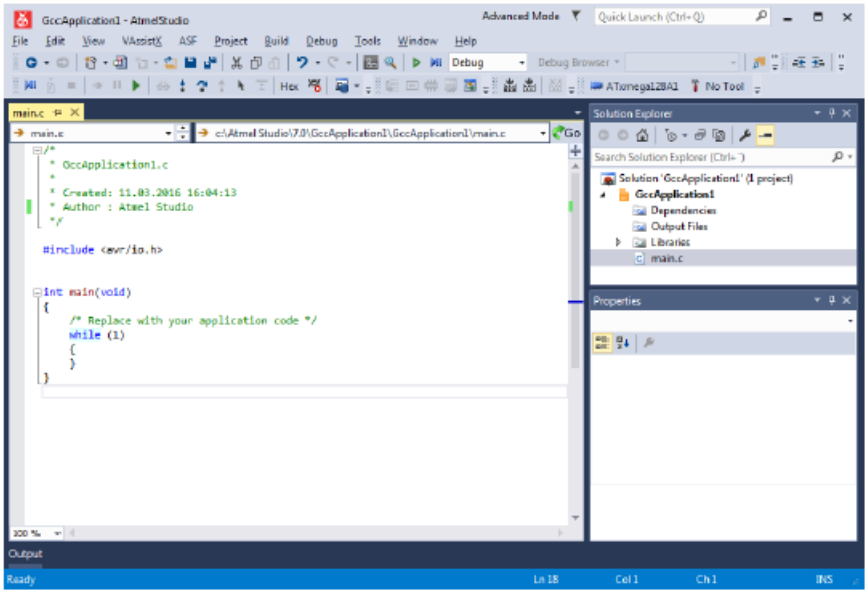



Figura: Atmel Studio tela de edição de código

As principais características desta ferramenta para este projeto são:

- Editor de código para linguagem C/C++ e Assembly
- Atmel Software Framework (ASF) com códigos exemplos e que permite a criação de aplicações modulares
- Ferramentas de debug e testes
- ARM GCC Toolchain
- Compatível com sistemas operacionais Windows XP, Windows Vista, / Windows 7/ 8

## 5.1 Linguagem de Programação

Para o projeto do sistema de aquisição de dados foi utilizada a **linguagem de programação C** com compilador ARM GCC Toolchain incluído no ambiente de desenvolvimento Atmel Studio.

## 5.2 Sistema Operacional em Tempo Real (RTOS)

Para o desenvolvimento do firmware foi integrado o sistema operacional em tempo real **FreeRTOS**. O objetivo de utilizar este RTOS é que, além de gratuito, ele possui uma pilha TCP/IP que é utilizada para a comunicação com o HOMS via interface Ethernet.

A versão do FreeRTOS integrada no sistema é o release 10.3.1 que foi liberada em 18 de Fevereiro de 2020. Novos releases podem ser acompanhados através do link https://github.com/FreeRTOS/FreeRTOS

## 6.0 Arquitetura do Firmware

O firmware do sistema de aquisição de dados é composto dos seguintes componentes:

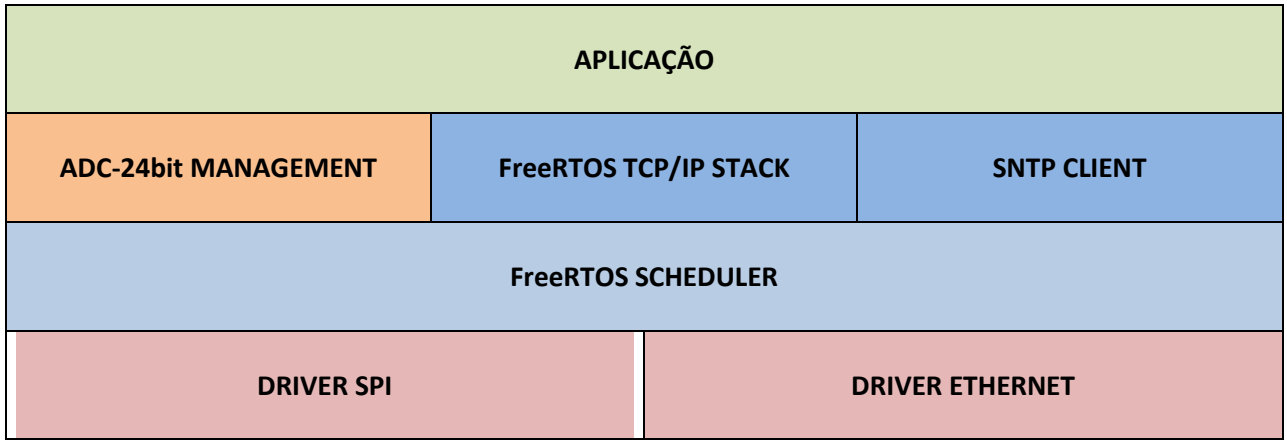



Figura: Arquitetura do Firmware do Sistema de Aquisição de Dados

Onde:

- DRIVER SPI
  Módulo responsável em controlar e executar a comunicação via interface SPI com a placa de aquisição de dados EVAL-AD7770 ADC 24-bit.

- DRIVER ETHERNET
  Módulo responsável em controlar e executar a comunicação via interface Ethernet com o HOMS.

- FreeRTOS SCHEDULER
  Módulo do FreeRTOS responsável em gerenciar as tarefas do firmware e realizar a comunicação de dados entre estas tarefas.

- FreeRTOS TCP/IP STACK

  Módulo do FreeRTOS que permite transmitir e receber dados via protocolo de comunicação TCP/IP.

- SNTP CLIENT

  Simple Network Time Protocol Client (SNTP) utilizado para sincronizar o relógio de tempo real do microcontrolador Cortex M4  ATSAM4E. O SNTP é menos complexo que o NTP e é recomendado para ser utilizado em aplicações embarcadas onde não é necessário a implementação de todas as funcionalidades do NTP.

- ADC 24-bit MANAGEMENT

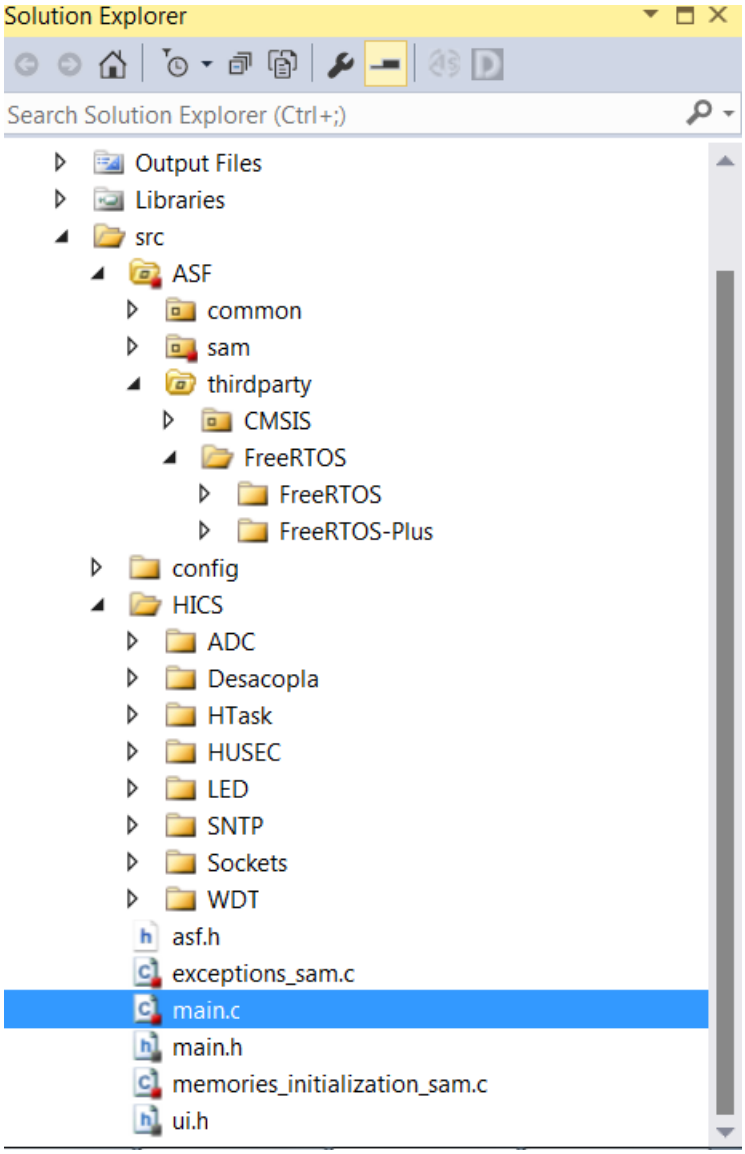
  Módulo responsável em inicializar, configurar e fazer as leituras da placa de aquisição de dados EVAL-AD7770 ADC 24-bit.

- APLICAÇÃO

  Módulo responsável em armazenar as leituras da placa de aquisição de dados, registrar o timestamp destas leituras e transmitir estas informações para o HOMS.

## 6.1 Estrutura das Pastas do Código Fonte

O código fonte do projeto está organizado nas seguintes pastas:




Onde as principais pastas são:

| PASTA | DESCRIÇÃO |
|---|---|
| ASF | Atmel Software Framework - código fonte dos drivers de suporte da placa SAM4E Xplained Pro |
| FreeRTOS | Código fonte do FreeRTOS |

| | |
|---|---|
| FreeRTOS-PLUS | Código fonte do stack TCP/IP do FreeRTOS |
| Config | Arquivos de configuração do sistema como módulos habilitados, clock, Ethernet e FreeRTOS |
| ADC | Código fonte das funções de controle, leitura e escrita da placa ADC |
| Desacopla | Código fonte da função Desacopla Placas |
| HTASK | Tasks do HICS |
| HUSEC | Código fonte do HUSEC |
| LED | Código fonte do modo de sinalização do LED da placa |
| SNTP | Código fonte da task SNTP |
| Sockets | Código fonte dos sockets |
| WDT | Gerenciamento do watchdog timer |
| main.c | Arquivo principal |

## 6.2 LED sinalizador

O **Led 0** da placa SAM4E sinaliza o estado de funcionamento do sistema. Os estados possíveis são:

| LED | PROCEDIMENTO |
|---|---|
| APAGADO | Placa SAM4E desligada |
| PISCANDO A CADA 250ms (4Hz) | Placa SAM4E ligada e no modo normal de funcionamento |
| PISCANDO A CADA 750ms (1,33Hz) | Processo de desacoplamento iniciado e sockets sendo fechados |
| PISCANDO A CADA 50ms (20Hz) | Placas SAM4E e EVAL-AD7770 desacopladas |

## 6.3 Função para Desacoplar as Placas

As placas EVAL-AD7770 e SAM4E Xplained Pro estão interligadas por sinais de controle e barramento de comunicação SPI conforme ilustrado na figura abaixo:

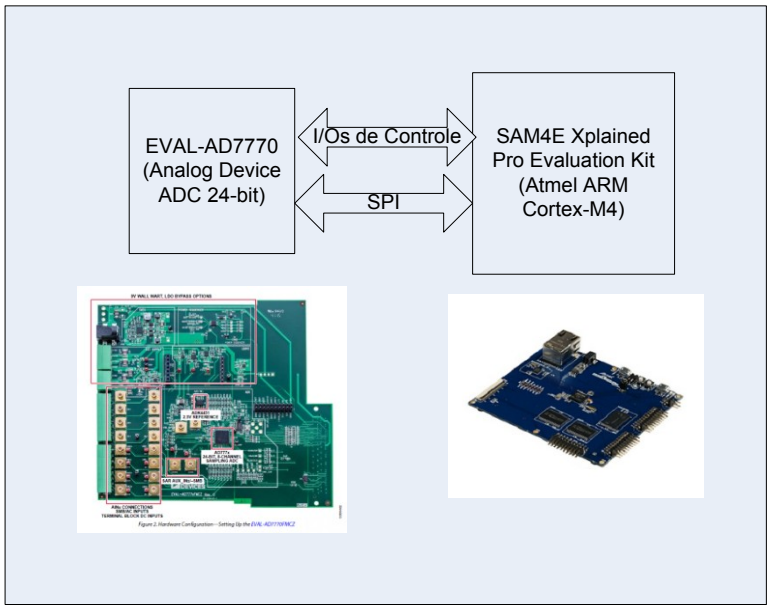



Figura: Conexões entre Placas AD7770 e SAM4E

Para assegurar o desacoplamento elétrico entre as placas ao manipulá-las, foi implementado o seguinte procedimento:

| PASSO | PROCEDIMENTO |
|---|---|
| 1 | Placas interligadas e sistema ligado e funcionando normalmente (led piscando a cada 250ms) |
| 2 | Manter o switch 0 (SW0) da placa SAM4E pressionado por mais que 3 segundos até que o led mude o seu padrão de piscar |
| 3 | Caso o led comece a piscar lentamente (750ms), aguardar pois os sockets estão sendo fechados |
| 4 | Quando o led começar a piscar rápido (50ms) as placas podem ser desligadas |

## 6.4 Timestamp

O sistema possui 2 timestamps que são transmitidos para o HOMS:

| TIMESTAMP | DESCRIÇÃO |
|---|---|
| Unix time ("Epoch") | Contador de segundos desde 01-01-70 00:00:00 |
| HUSEC | Número de centenas de microsegundos (=0,1 milisegundos) do dia de observação |

## 6.5 Endereçamento de Rede

O endereçamento de rede da placa SAM4E está definido no arquivo main.c e pode ser alterado via compilação. Os valores definidos são:

| PARÂMETRO | ENDEREÇO PADRÃO | NOME DA CONSTANTE |
|---|---|---|
| IP ADDRESS | 192.168.1.201 | ucIPAddress |
| MASK | 255.255.255.0 | ucNetMask |
| GATEWAY | 192.168.1.1 | ucGatewayAddress |
| MAS ADDRESS | 00.11.22.33.44.55 | ucMACAddress |

## 6.6 Sockets

Os sockets do sistema estão definidos no arquivo sockec.c e podem ser alterados via compilação. Os valores definidos são:

| SOCKET | TCP/UDP | DESCRIÇÃO | ENDEREÇO PADRÃO DO SERVIDOR |
|---|---|---|---|
| ADC | TCP | Socket para transmissão do HICS para o HOMS das leituras da placa de aquisição de dados | 192.168.1.100 :: 15001 |
| SNTP | UDP | Socket SNTP | 192.168.1.100 :: 123 |

## 6.7 Amostragem da Placa de Aquisição de Dados

A placa de aquisição de dados EVAL-AD7770 é amostrada a cada 1ms.

Os dados amostrados são copiados para um buffer circular na memória SRAM externa da placa SAM4E Xplained Pro.

A capacidade de armazenamento deste buffer circular é de até 10.000 amostras.

Os dados são armazenados conforme a seguinte estrutura:

| CAMPO | DADO | FORMATO | DESCRIÇÃO |
|---|---|---|---|
| 1 | ulSample | uint32_t | Número sequencial das amostras |
| 2 | ulTimestamp_sec | uint32_t | Timestamp em segundos (Unix Epoch) |
| 3 | usTimestamp_ms | uint16_t | Timestamp em milesegundos |
| 4 | u64husec | uint64_t | HUSEC |
| 5 | rawGolay[4] | uint8_t | Buffer 4 bytes da amostragem ADC da Célula de Golay |
| 6 | rawPS[4] | uint8_t | Buffer 4 bytes da amostragem ADC da Fonte de Alimentação |

## 6.8 Protocolo de Comunicação

O protocolo de comunicação para a transmissão dos dados amostrados da placa de aquisição EVAL-AD7770 para o HOMS utiliza o socket ADC (ver item 6.6 acima).

Os dados de cada amostragem são transmitidos por linhas, sendo que em cada linha temos os 6 campos da estrutura comentada no item 6.7 acima

Exemplo de transmissão dos dados:

| | CAMPO | | | | | |
|---|---|---|---|---|---|---|
| **LINHA** | **UlSample** | **ulTimestamp_sec** | **usTimestamp_ms** | **u64husec** | **rawGolay[4]** | **rawPS[4]** |
| 1 | 00004E61 | 5F2AFAEA | 034A | 0000000027BC8D8D | B2FFFF88 | 83FFFFEB |
| 2 | 00004E62 | 5F2AFAEA | 034B | 0000000027BC8D97 | BFFFFF92 | 83FFFFE4 |
| 3 | 00004E63 | 5F2AFAEA | 034C | 0000000027BC8DA1 | B4FFFF9B | 85FFFFEC |
| … | . . . | . . . | . . . | . . . | . . . | . . . |
| … | . . . | . . . | . . . | . . . | . . . | . . . |

Os dados são transmitidos no formato ASCII-hexadecimal. Por exemplo, supondo o número 165 decimal, que tem representação hexadecimal 0xA5, este número será transmitido como dois bytes ASCII-hexadecimal: 0x41 e 0x35, onde 0x41 é a representação ASCII-hexadecimal da letra “A” e 0x35 é a representação ASCII-hexadecimal do número “5“.

Os dados da célula de Golay (rawGolay[4]) e da Fonte de Alimentação (rawPS[4]) estão no formato lido do conversor AD7770.

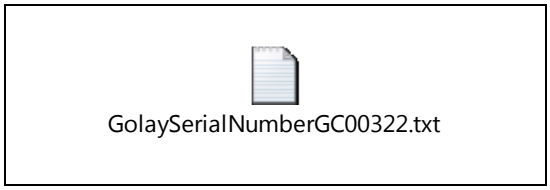
Por exemplo, para converter o valor rawGolay B2FFFF88 para decimal deve-se aplicar o seguinte procedimento:

1. Cortar o byte mais significativo: (B2)FFFF88
2. Converter o número hexadecimal FFFF88 para decimal (bit mais significativo sinaliza o sinal, se 1 o número é negativo, se 0 o número é positivo).
   Portanto neste exemplo o número FFFF88 hexadecimal é igual a -120 decimal

## 6.9 Exemplo de Arquivo de Dados

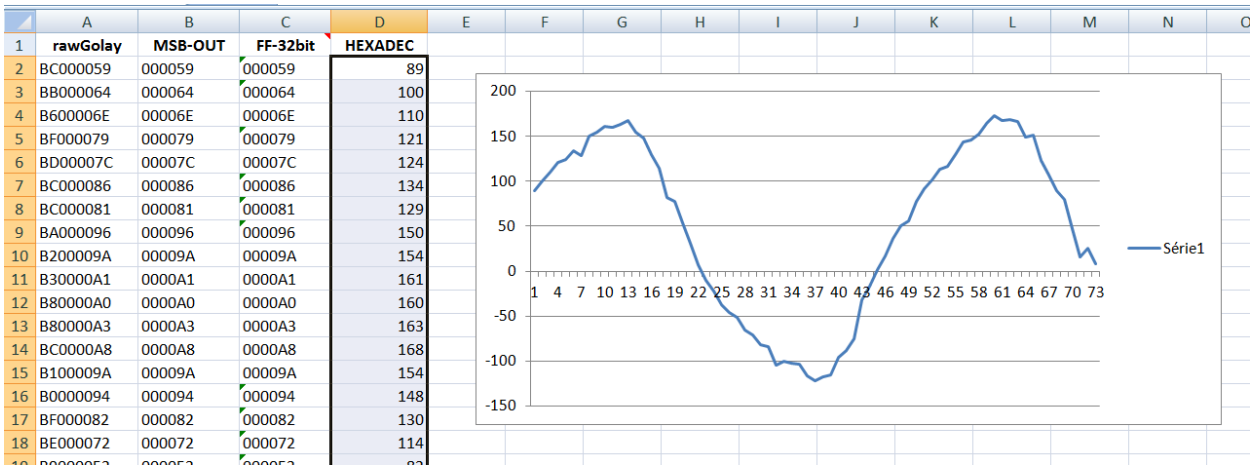
Abaixo segue exemplo de um arquivo de dados gerado a partir da transmissão do socket ADC para um servidor Netcat.

A linha de comando Netcat para gerar este arquivo foi: "ncat -v -l -p 15001 -o GolaySerialNumberGC00322.txt"




Arquivo: GolaySerialNumberGC00322.txt

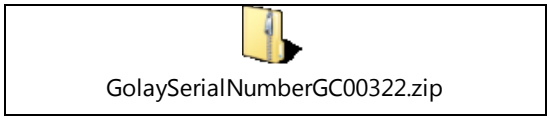
Abaixo segue um exemplo de gráfico convertendo os dados da Célula de Golay conforme mencionado no item 6.8 acima



| | A | B | C | D |
|---|---|---|---|---|
| 1 | rawGolay | MSB-OUT | FF-32bit | HEXADEC |
| 2 | BC000059 | 000059 | 000059 | 89 |
| 3 | BB000064 | 000064 | 000064 | 100 |
| 4 | B600006E | 00006E | 00006E | 110 |
| 5 | BF000079 | 000079 | 000079 | 121 |
| 6 | BD00007C | 00007C | 00007C | 124 |
| 7 | BC000086 | 000086 | 000086 | 134 |
| 8 | BC000081 | 000081 | 000081 | 129 |
| 9 | BA000096 | 000096 | 000096 | 150 |
| 10 | B200009A | 00009A | 00009A | 154 |
| 11 | B30000A1 | 0000A1 | 0000A1 | 161 |
| 12 | B80000A0 | 0000A0 | 0000A0 | 160 |
| 13 | B80000A3 | 0000A3 | 0000A3 | 163 |
| 14 | BC0000A8 | 0000A8 | 0000A8 | 168 |
| 15 | B100009A | 00009A | 00009A | 154 |
| 16 | B0000094 | 000094 | 000094 | 148 |
| 17 | BF000082 | 000082 | 000082 | 130 |
| 18 | BE000072 | 000072 | 000072 | 114 |



O arquivo utilizado para a geração do gráfico encontra-se abaixo:




Arquivo: GolaySerialNumberGC00322.zip

## 6.10 Releases do Firmware

| RELEASE | DATA | DESCRIÇÃO |
|---|---|---|
| 1.00 | 26/June/2020 | • Monitoramento do funcionamento do firmware e implementação de watchdogs (hardware e software)<br><br>• Buffer circular na memória SRAM externa<br>Capacidade: 10.000 amostras<br><br>• Transmissão dos pacotes via TCP ao invés de UDP<br>- Pacotes transmitidos como ASCII-HEX<br>- Campos (separados por espaço):<br>==> ulSample ulTimestamp_sec usTimestamp_ms u64husec rawGolay rawPowerSupply<br>==> Exemplo: 0018DE8F 5EF1150A 015D 000000002C071C19 B3000035 89FFFFE8<br><br>• Função para desacoplar placas antes de desligar alimentação<br>Procedimento:<br>1. Led piscando normal (250ms)<br>2. Manter switch 0 pressionado por mais que 3 segundos até led mudar padrão de piscar<br>3. Se led começar a piscar lentamente (750ms) aguardar (placa está desacoplando - desligamento suave - fechando sockets)<br>4. Quando led começar a piscar rápido (50ms) as placas podem ser desligadas<br><br>• Monitoramento de ruído na fonte de alimentação<br>Amostragem e transmissão do canal AD(0) junto com os pacotes de dados da Célula de Golay |
| 00.03 | 02/October/2020 | • Implementado SNTP client<br><br>• Implementado Timestamp HICS com SEGUNDOS, MILESSEGUNDOS E HUSECS<br><br>• Implementado socket UDP para transmitir buffer das amostras ADC com Timestamp<br>==> Dados transmitidos: NUMBER_OF_SAMPLE TIMESTAMP_SEC TIMESTAMP_MS HUSECS SAMPLE |
| 00.02 | 03/September/2019 | • Implementado Driver SPI<br><br>• Implementado ADC-24bit MANAGEMENT |

| | | |
|---|---|---|
| | | ==> Implementado leitura e escrita dos registros de configuração do AD7770<br>==> AD7770 configurado para:<br>- POWERMODE = High resolution<br>- DECIMATION RATE = 2048 (Output Data Rate ODR=1KHz)<br>- ENABLE SPI slave mode to read back ADC on SDO<br>==>Implementado leitura das amostras do canal 3 do AD via interrupção a cada 1ms<br>- AD7770 DRDY signal trigger an interrupt on rising edge<br>- DRDY gera int a cada 1ms (DECIMATION RATE = 2048 (ODR = 1KHz))<br>==> Implementado teste em ADC_task() para imprimir no debug 1000 amostras do canal 3 a cada 5s em formato decimal<br><br>• configASSERT habilitado para debug do FreeRTOS<br><br>• Code OPTIMIZATION passado de nível 1 para nível 0 (não otimizar) para facilitar depuração passo a passo |
| 00.01 | 28/June/2019 | • Rotinas de inicializacao da placa Atmel SAM4E-Xplained-Pro<br>• Freertos-10.0.0<br>• FreeRTOS_Plus_TCP<br>• Tick 1ms |

## Controle de Versões do Documento

| Versão | Data | Autor | Descrição |
|---|---|---|---|
| 2.0 | 2020-08-24 | Márcio Zaquela | Relatório Final |
| 1.0 | 2019-05-14 | Márcio Zaquela | Primeira Versão |