# Датчик усилия (strain gauge)

Используется SIKA FTCE

Maximum capacity (Emax) 300 / 500 kg / 1 / 2.5 / 5 / 7.5 / 10 / 20 t

Rated characteristic value (Cnom) 2 mV / V

Input resistance 420 ± 20 Ω

Output resistance 350 ± 2 Ω

Insulation resistance > 5 GΩ

Zero signal tolerance <1 % of Cnom

Supply voltage 1...15 V (typically 10 V)

Operating voltage 1...18 V

Бюджет для ADS1232:

номинал при 5В и 2мВ/В это 10мВ

при G=128 входной диапазон = +/-20мВ

2mV/V\*5V\*128 = 1280mV

Мысли:

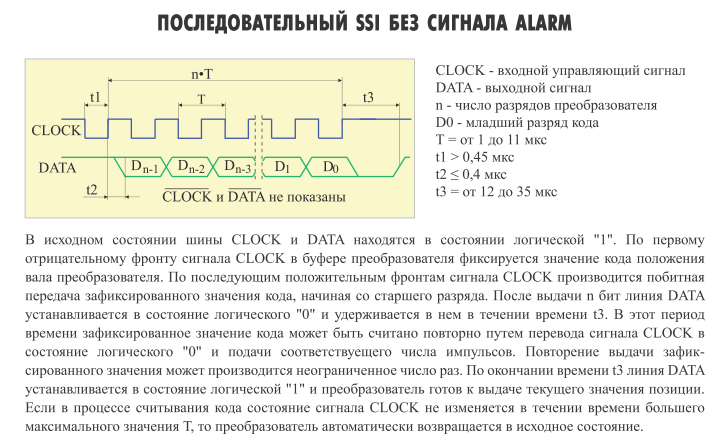
\* следить за уходом нуля

например перед включением всегда делать холостой проход и сравнивать с запомненным заводским значением, если отличается на 5% то АВАРИЯ

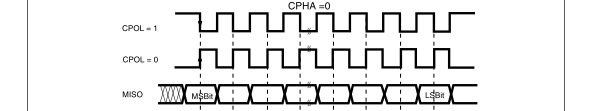
Измерение сопротивления проводов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RED | BLACK | YELLOW | WHITE |
| RED |  | 453 | 312 | 312 |
| BLACK |  |  | 312 | 312 |
| YELLOW |  |  |  | 350 |
| WHITE |  |  |  |  |

# Абсолюдный энкодер



T = 1..11us => f = 91kHz...1MHz



/\*

clk 1110101010

dat 1111xxyyzz

\*/

# Серво-контроллер

Shielded twisted-pair STP, Cat.5 cables must be used for cabling

http://www.ulmart.ru/goods/335173

экранированный патч-корд кат.6, NIKOMAX

modbus port 502

The following Modbus commands are supported:

read holding registers 3 Read process data (word by word)

write multiple registers 16 Write process data (word by word)

As a Modbus/TCP user, the motor controller can be reached via the same IP address as also used by FCT.

http://www.simplymodbus.ca/TCP.htm

http://www.mtrele.ru/files/documents/modbus.pdf

**PDU = protocol data unit**

**MBAP header = Modbus Application Header**

## Read Holding Registers (FC=03) Request

|  |  |
| --- | --- |
|  | **00 01**  **Transaction Identifier: 2 bytes set by the Client to uniquely identify each request. These bytes are echoed by the Server since its responses may not be received in the same order as the requests.** |
|  | **00 00**  **Protocol Identifier: 2 bytes set by the Client, always 0000** |
|  | **00 06**  **Length:  2 bytes identifying the number of bytes in the message to follow.** |
| 11  The SlaveID Address | **11**  **Unit Identifier: 1 byte set by the Client and echoed by the Server for identification of a remote slave connected on a serial line or on other buses.** |
| **03**  **The Function Code 3** |  |
| **006B**  **The Data Address of the first register requested.**  **( 006B hex = 107 , + 40001 offset = input #40108 )** |  |
| **0003**  **The total number of registers requested.**  **(read 3 registers 40108 to 40110)** |  |
| 7687  The CRC |  |

## Read Holding Registers (FC=03) Response

|  |  |
| --- | --- |
|  | **00 01** |
|  | **00 00** |
|  | **00 09** |
| 11  The Slave Address | **11** |
| **03**  **The Function Code 3** |  |
| **06**  **The number of data bytes to follow**  **(3 registers x 2 bytes each = 6 bytes)** |  |
| **AE41:**  **The contents of register 40108** |  |
| **5652:**  **The contents of register 40109** |  |
| **4340:**  **The contents of register 40110** |  |
| 49AD:  The CRC | 41 bytes total |

## Preset Multiple Registers (FC=16) Request

|  |  |
| --- | --- |
|  | **00 01** |
|  | **00 00** |
|  | **00 0F**  **number of bytes in the message to follow** |
| 11  The Slave Address | **11** |
| **10**  **The Function Code 16** | **10** |
| **0001**  **The Data Address of the first register.**  **( 0001 hex = 1 , + 40001 offset = register #40002 )** | **00 00** |
| **0002**  **The number of registers to write** | **00 04** |
| **04**  **The number of data bytes to follow**  **(2 registers x 2 bytes each = 4 bytes)** | **08** |
| **000A**  **The value to write to register 40002** | **11 22** |
| **0102**  **The value to write to register 40003** | **33 44** |
|  | **55 66** |
|  | **77 88** |
| C6F0  The CRC |  |

## Preset Multiple Registers (FC=16) Response

|  |  |
| --- | --- |
|  | **00 01** |
|  | **00 00** |
|  | **00 06**  **number of bytes in the message to follow** |
| 11  The Slave Address | **11** |
| **10**  **The Function Code 16** | **10** |
| **0001**  **The Data Address of the first register.**  **(# 40002 - 40001 = 1 )** | **00 00** |
| **0002**  **The number of registers written.** | **00 04** |
| 1298  The CRC | 12 bytes total |

## FHPP

Record Torque Limitation CMMP 418 1 ... 250 uint32

Torque/current current limitation in Profile Position mode in mNm.

Direct Mode Velocity Torque Limit CMMP 565 1 uint32

Torque limitation during direct Profile Velocity mode in unit of torque (mNm).

Homing Max. Torque CMMP 1015 1 uint8

Max. torque during homing.

Specified as a multiple of the rated torque in % (see PNU 1036).

The maximum permitted torque (via current limiting) during homing. If this

value is reached, the drive identifies the stop (REF) and travels to the axis

zero point.

Motor Rated Torque All 1036 1 uint32

The motor’s rated torque in 0.001 Nm.

# Биомеханика

Типы мышечных сокращений

|  |  |
| --- | --- |
| Концентрическое  Из B в A | укорачивание мышцы  жим штанги - фаза поднимания |
| Эксцентрическое  Из A в B | удлинении мышцы  жим штанги - фаза опускания |
| Изометрическое | длина мышцы не изменяется |

Математическая модель

|  |
| --- |
| Ход механизма условно разбивается на шкалу 0..100%  A =13.4% - начало эксцентрического движения, конец концентрического движения (жим штанги - верх)  B =79.3% - конец эксцентрического движения, начало концентрического движения (жим штанги - низ) |
|  |
|  |
|  |
|  |

# USS

# ILDA

# Актуаторы

Разъем PY07-7

1 синий BLUE

2 коричнеый BROWN

3

4 белый VIOLET

5 желтый WHITE

6 черный BLACK

7

Клеммники на плате

№1 - сигнальный

1 GND черный

2 24VDC желтый

3 ANALOG белый

4 GND

5 24VDC

6 SIG

…

№2 - силовой

1 + коричневый

2 - синий

3 +

4 -

# §MFRC522

https://github.com/miguelbalboa/rfid

|  |
| --- |
| #include <SPI.h>  #include <MFRC522.h>  #define SS\_PIN 10  #define RST\_PIN 9  MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.  void setup() {  SPI.begin(); // Init SPI bus  mfrc522.PCD\_Init(); // Init MFRC522 card  Serial.println("Scan PICC to see UID and type...");  }  void loop() {  // Look for new cards  if ( ! mfrc522.PICC\_IsNewCardPresent()) {  return;  }  // Select one of the cards  if ( ! mfrc522.PICC\_ReadCardSerial()) {  return;  }  // Dump debug info about the card. PICC\_HaltA() is automatically called.  mfrc522.PICC\_DumpToSerial(&(mfrc522.uid));  } |