بسم الله الرحمن الرحيم

MEGBI Antibiotics/Antibodies Production Pilot Plant (MEGBI-APP)- 5th Project Report (2017)

- Completing integration of MEGBI-APP test rig (valves, automation)
- Chemical Process Simulation of MEGBI-APP
- Study for monoclonal antibodies production

Authors:

Samir Mourad, Ahmad Trad, Asia Mourad, Rami Nassouh, Razan Kalaoun, Fatima Antar

Initial Document: 9 June 2017

Last update/آخر تعديل: 24 Dezember 2017

مركـز أبحـاث الشـرق الأوسـط للجينـات والتقنيـة البيولوجية

رأسنحاش – قضاء البترون- لبنان

Middle East Genetics and Biotechnology Institute (MEGBI)

A Member Institute of AECENAR

Main Road, Ras-Nhache, Batroun, Lebanon

http://www.aecenar.com/institutes/megbi

www.temo-ek.de

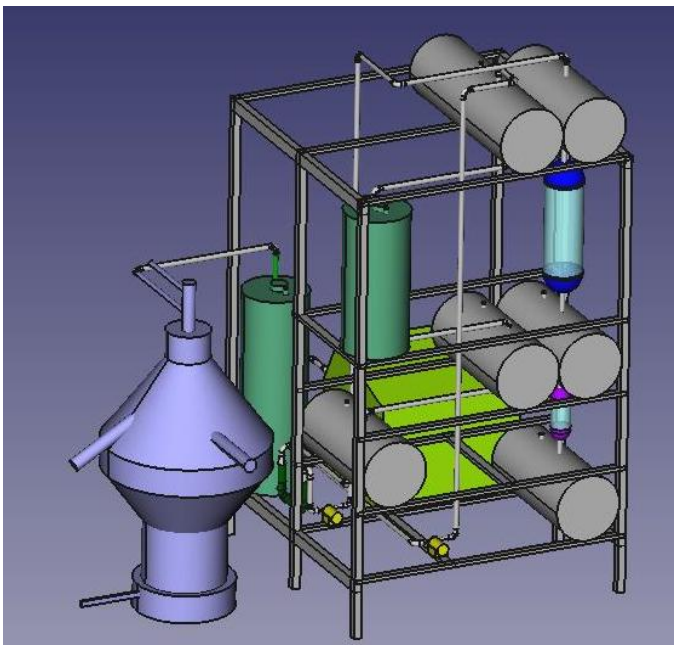
TEMO Biotechnology

TEMO e.K., Im Klingenbühl 2a, 69123 Heidelberg, Germany

الشركة اللبنانية الالمانية للبيوتكنولوجيا

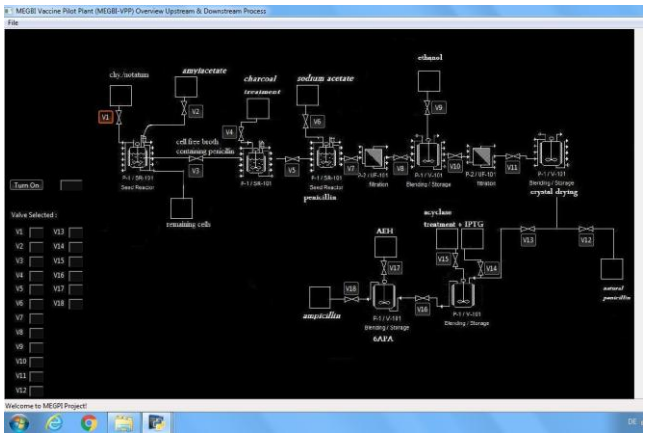
Lebanese-German Biotech

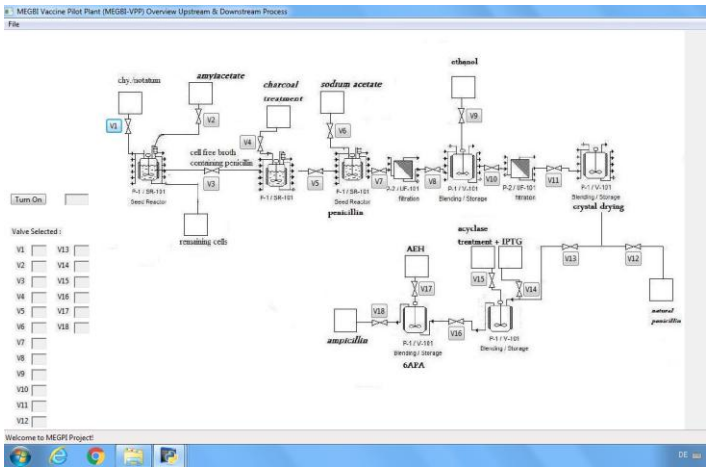
المضمون



060116MEGBI-VPP_Assembly_simplified_manufactured.FCStd

- Mechanical parts of minimal USP-DSP manufactored





Process Control System

05121621uhr_MEGBI-APP_automation.py

- Automation system, Graphical User Interface

1 إدارة المشروع (Project Management)

1.1 هدف /العمل (Project phase goal)

The goal is to install a pilot plant for producing semi-synthetic penicillin.

1.2 Budget Planning

From 3rd project report, Ch. 1.4: Azm Association (Hani Maulawi, Dr Dani Saaduddin, Dr Kifah Tout) visited AECENAR Center at Ras Nhache on 6th March 2015 and Business Plan 2 was discussed. Result (Status 17th March 2015): Azm wants a more **detailed business plan with detailed market strategy. This is to be done in 2018.**

1.3 Time Schedule / الجدول الزمني/

Originally planned: Nov/Dec 14: Financement and Concept Phase

Jan – June 15: Finishing of Development of MEGBI Vaccine Production Pilot Plant (MEGBI-VPP)

Actually: March-May 2016: Migration of specification to semi-synthetic penicillin plant, 2017: Completing of MEGBI-APP prototype

1.3.1 Timeplan Sep-Dec 2017

Timeplan MEGBI-APP Sep-Dec 2017

Task	Due Date	Responsible
.pcd files of every compound (about 20)	Sun, 10.9.17	Razan
Moving lab (chemicals, some devices) to Hamra Facility	Sun, 10.9.17	Samir
Documentation	Sun, 10.9.17	Razan
Beginning of lab work	Sa, 16.9.17	Razan, Mariam, Rayyan
Integration MEGBI-APP plant	October 17	CNC Lab, Samir

1.4 Costs for completing protoype for Ampicillin production

1.4.1 Alternative 1: Stepper Motor for automatic valves

Automatic valves					
	piece cost	#pieces	total pieces		
سكر	$4	18	$72		
stepper motor	$40	18	$720		
acessories motor	$10	18	$180		
		Total cost	**$972**		

1.4.2 Alternative 2: DC Motor for automatic valves

Metal DC Geared Motor - 12V 50RPM 9kg.cm rated torque, Price : 15.95$, Serial number : ACT0022.

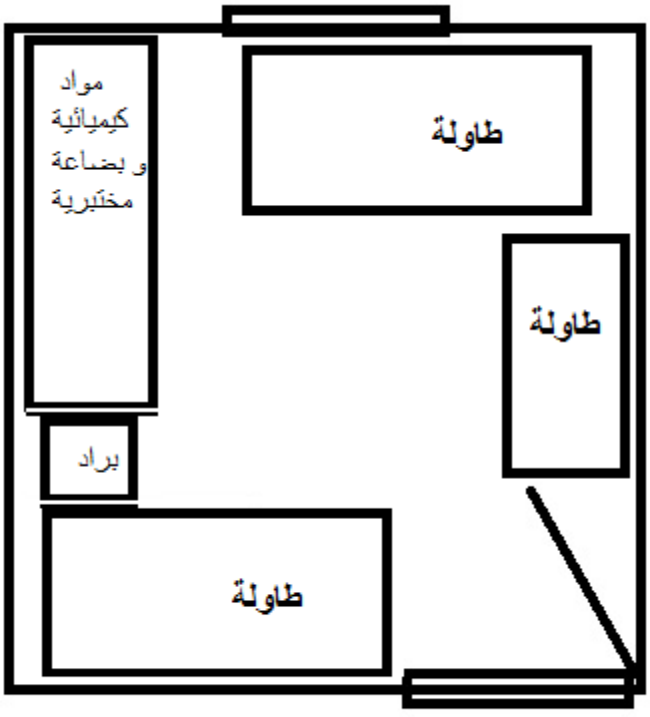
Automatic valves					
	piece cost	#pieces	total pieces		
سكر	$4	18	$72		
DC motor	$15	18	$270		
acessories motor	$10	18	$180		
		Total cost	**$522**		

1.4.3 Alternative 3: Low cost servo (see chapter 6)

Automatic valves					
	piece cost	#pieces	total pieces		
سكر	$4	18	$72		
low cost servo 9kg.cm	$8	18	$144		
acessories motor	$10	18	$180		
			$0		
		Total cost	**$396**		

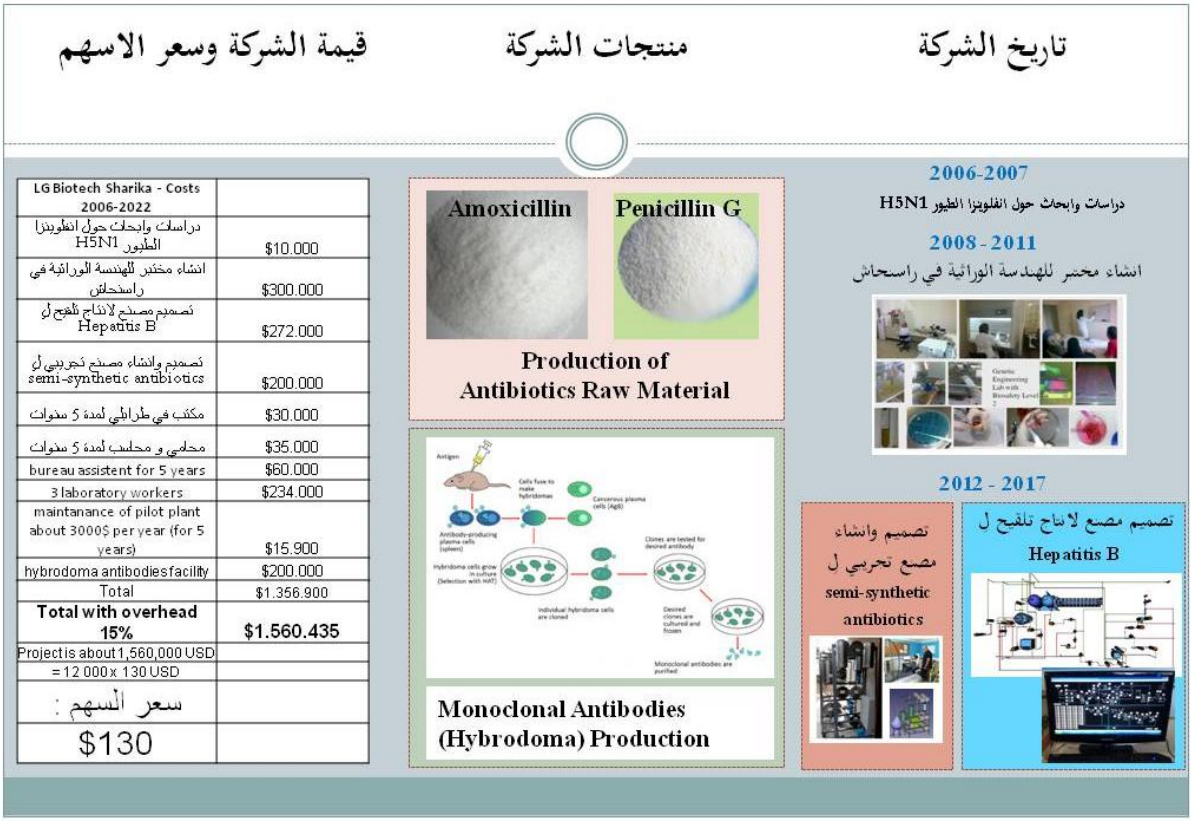
1.4.4 Lab in Ras Nhache (was planned in Sep 2017)

Lab Ras Nhache



2 Marketing

2.1 Brochure for LG Biotech



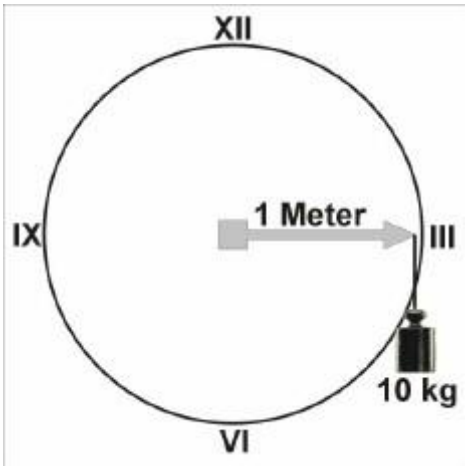
2.2 Feasibility Study for Antibiotics production, Vaciccine production, Monoclonal Antibodies production

To be done in LG Biotech

3 Basics

3.1 Torque at Stepper Motors and Servos

Wenn man an den Zeiger einer Turmuhr in der Stellung auf 3 Uhr ein Gewicht von 10 kg hängt, wirkt auf die Achse ein Drehmoment von 100 Nm (also 10000 Ncm). Ein Getriebemotor mit 100 Ncm könnte beispielsweise bei einem Hebel von 1 cm (an der Achse) noch 10 kg heben.



3.1.1 Product Example (from www.cnclablb.com)

	Metal Gear Servo TowerPro MG995 Servo - 9kg Price : 8$ Serial number : ACT0005

Description:

Modulation: Digital

Torque: 4.8V: 130.54 oz-in (**9.40 kg-cm**) 6.0V: 152.76 oz-in (11.00 kg-cm)

Speed: 4.8V: 0.20 sec/60° 6.0V: 0.16 sec/60°

Weight: 1.94 oz (55.0 g)

Dimensions:Length:1.60 in (40.7 mm)

Width:0.78 in (19.7 mm)

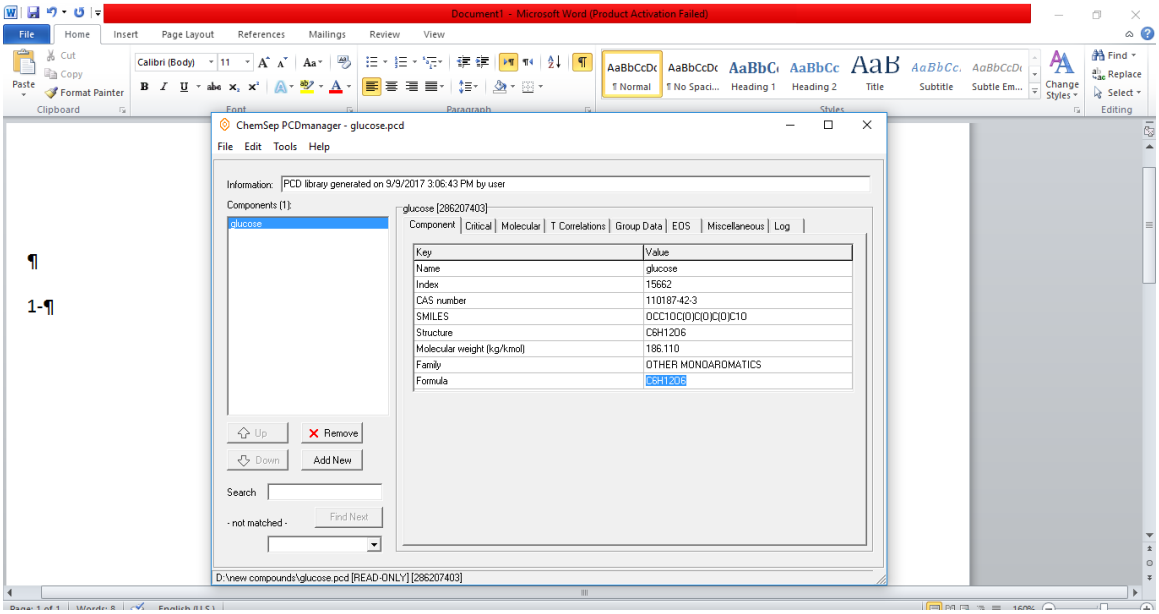
Height:1.69 in (42.9 mm)

3.2 Chemical Process Simulation with COCO

3.2.1 How to add new compounds with COCO

❖ Steps:

1- Open PCD manager



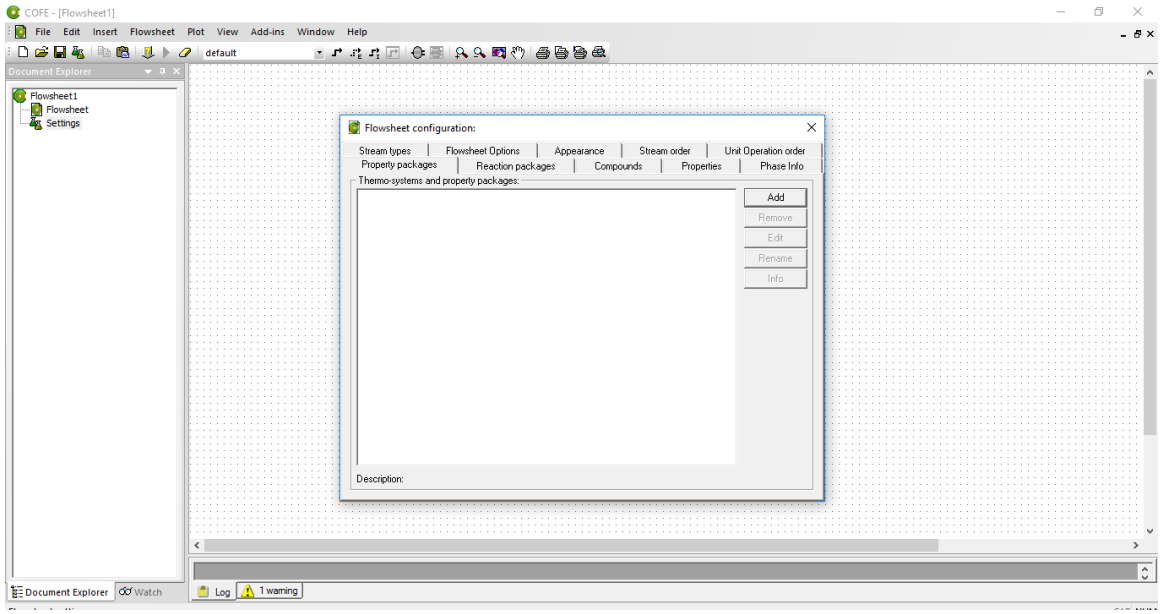
2- Press Add New

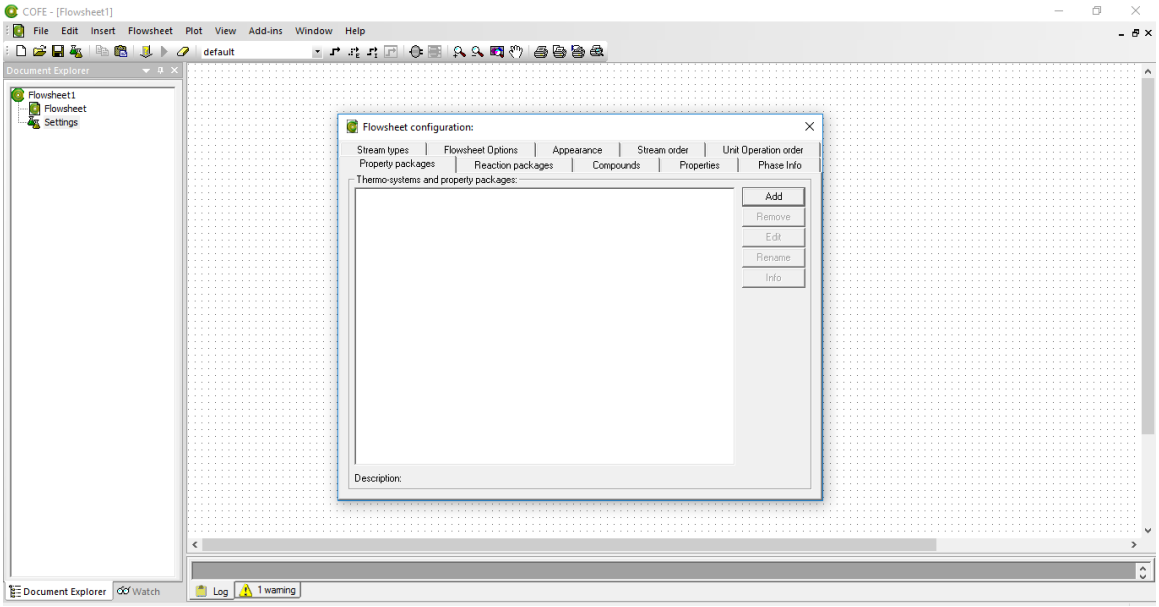
3- Enter compound's information

4- Save as in a file in local disk D

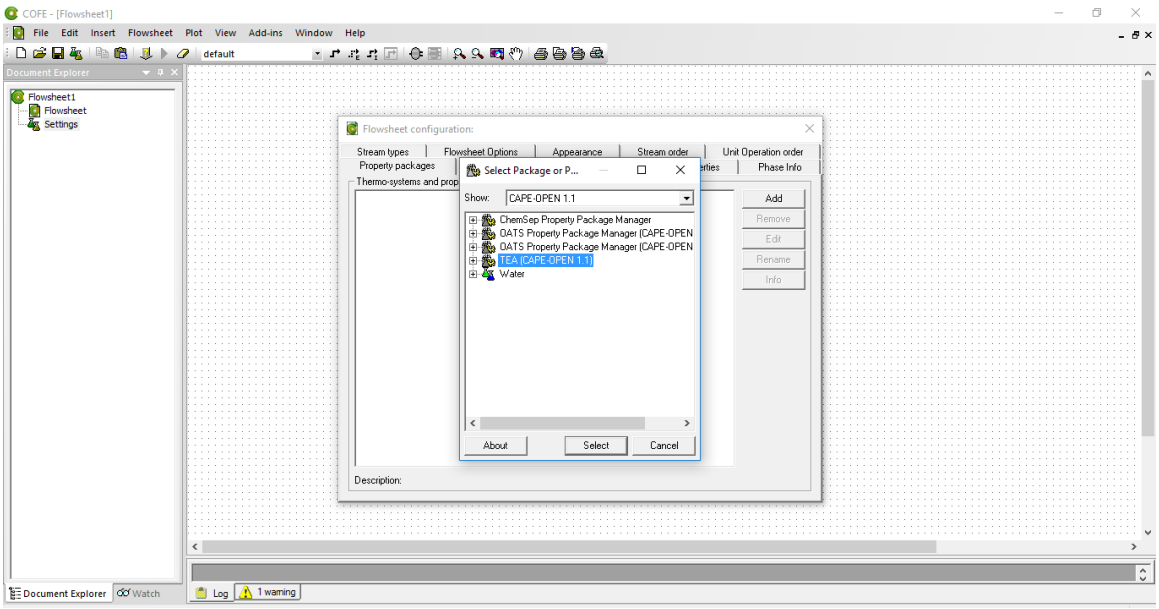
5- Open coco program

6- Press sittings(left) then press new

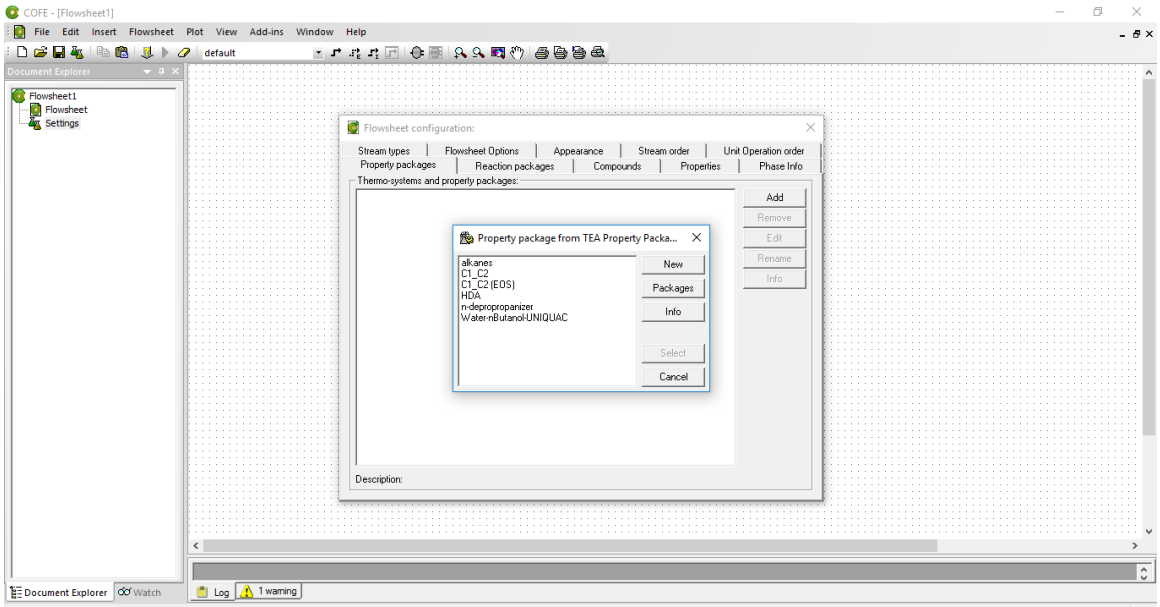




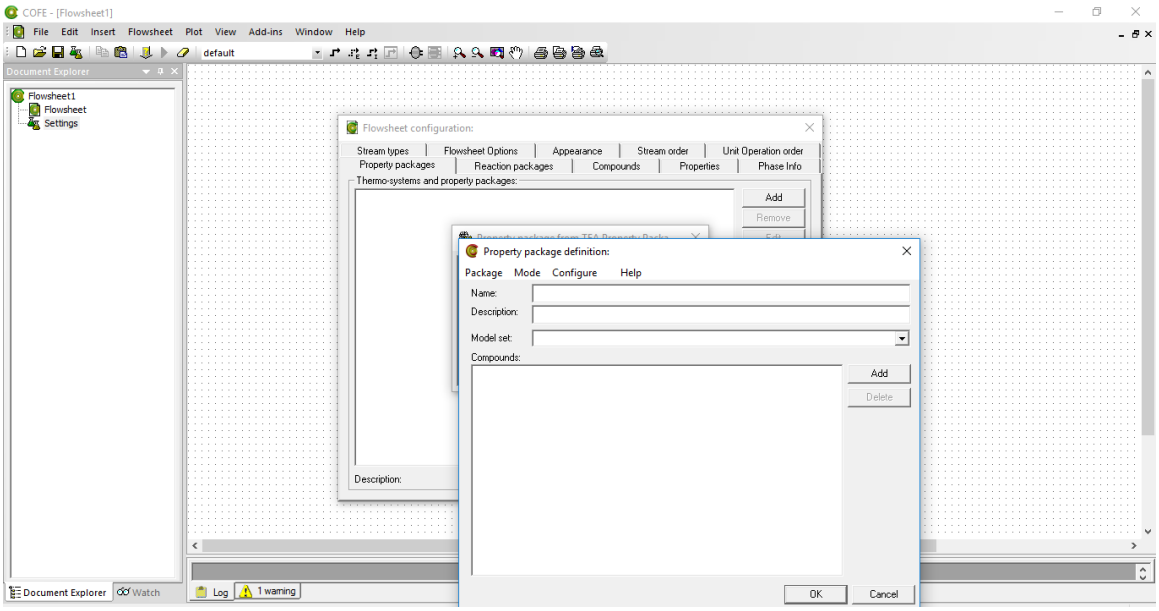
7-Select tea



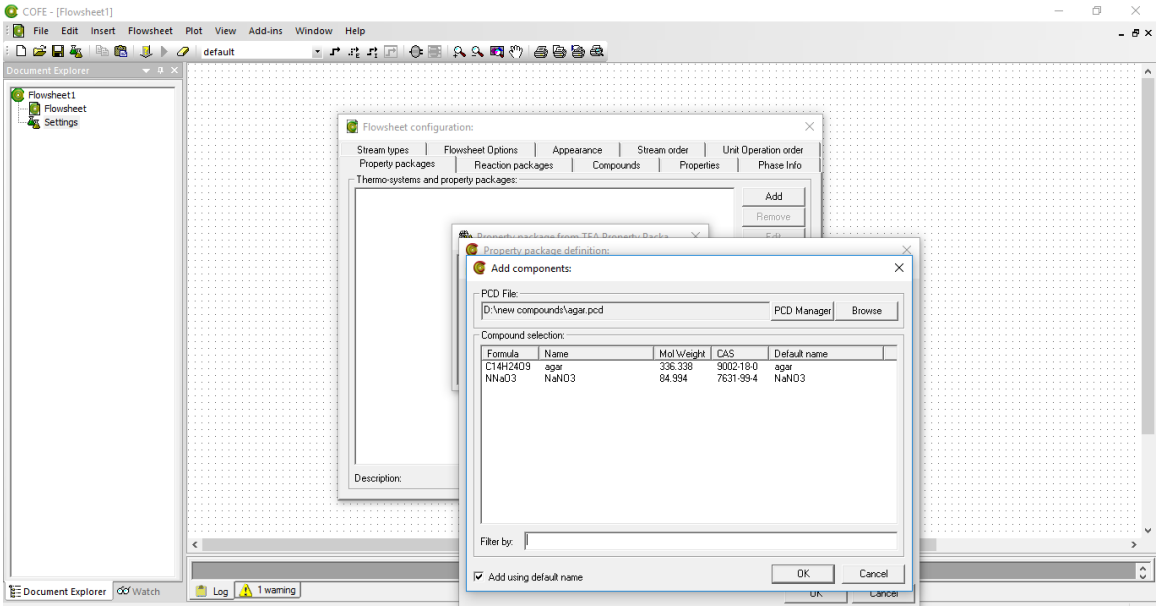
8-Then press new



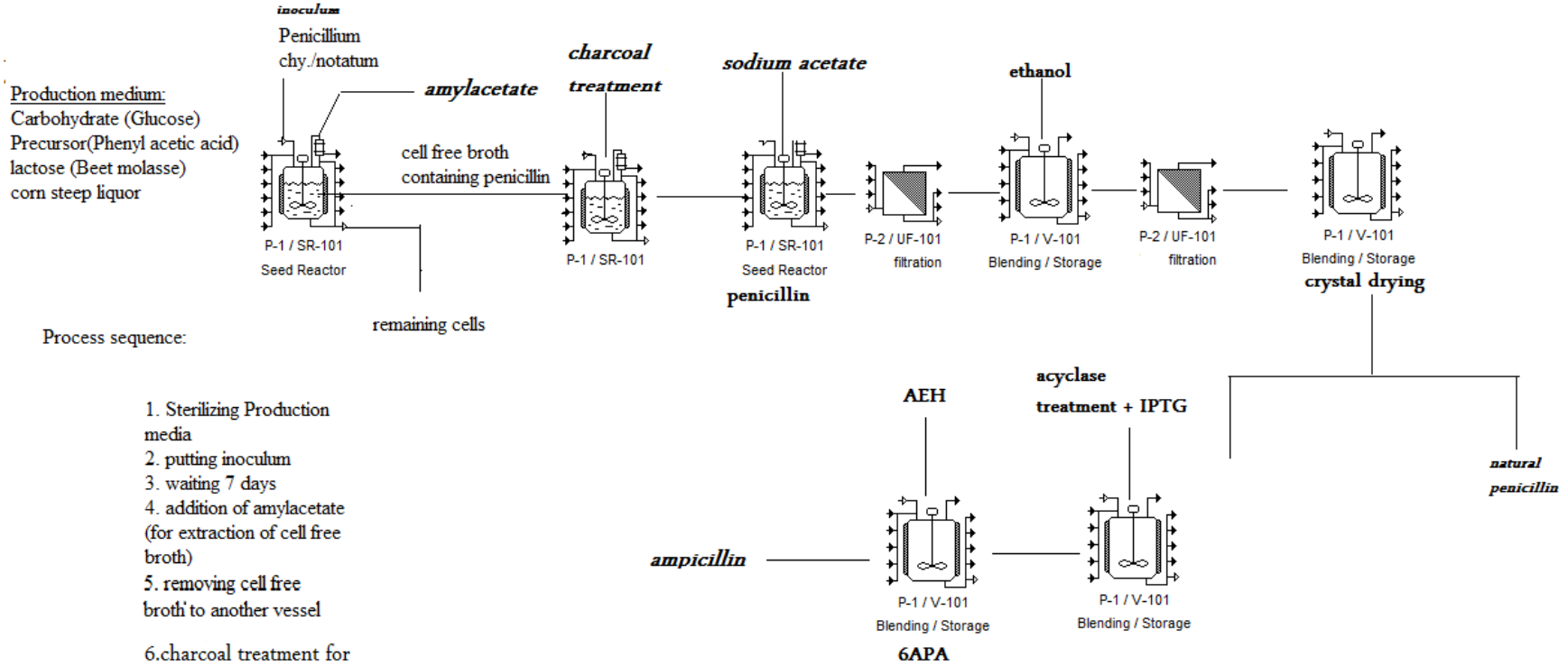
9-Press add



10-press browse (right)

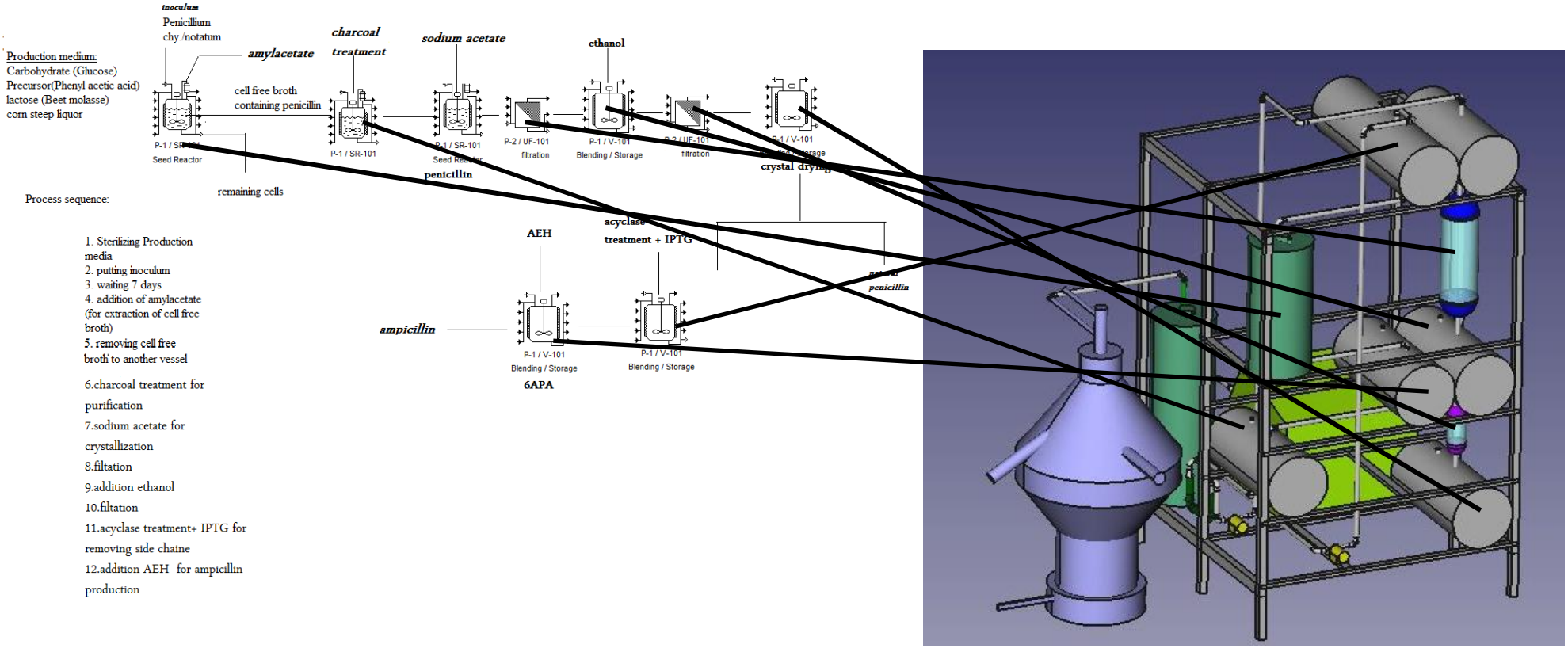


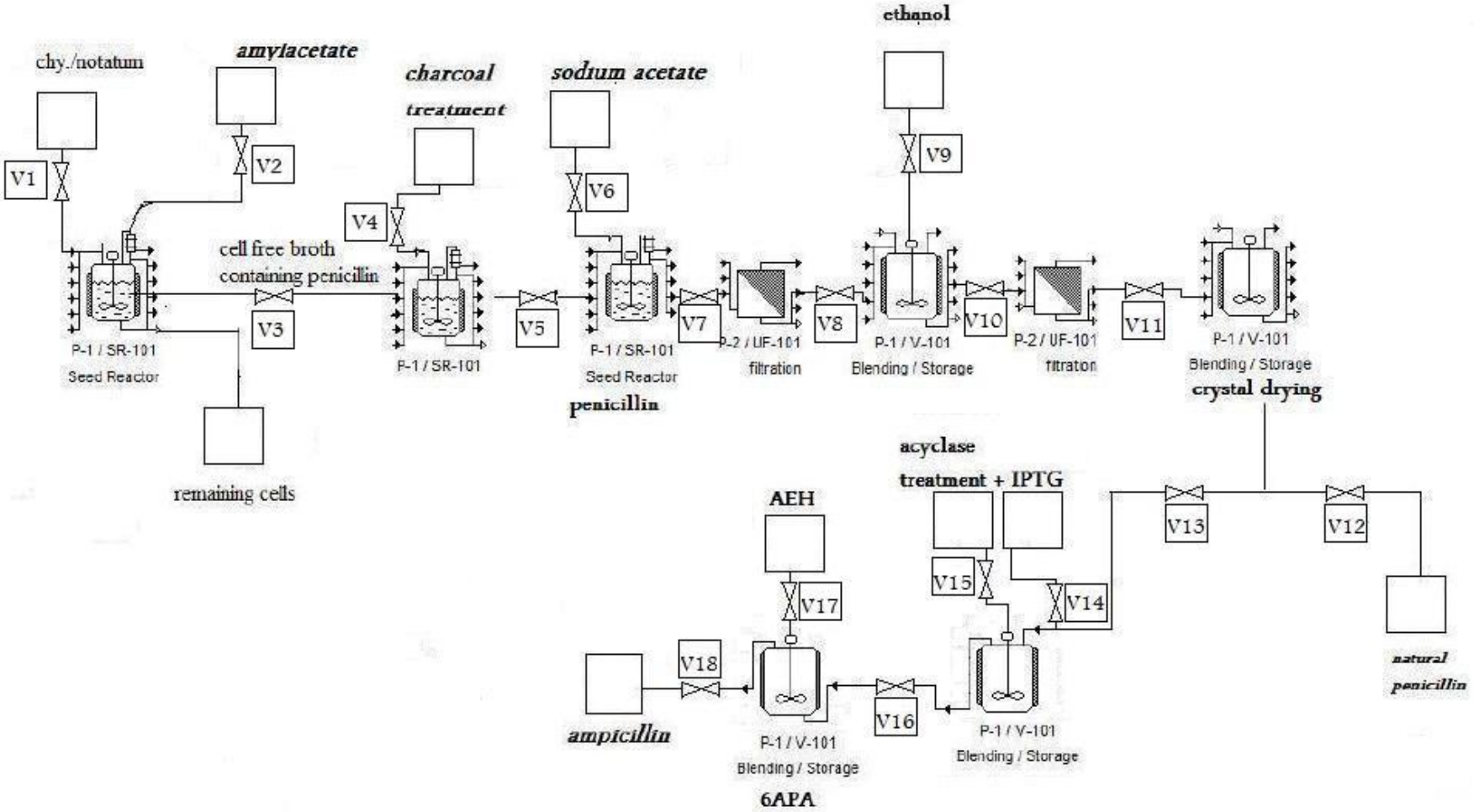
11-choose the compound you add it



Process sequence:

1. Sterilizing Production media
2. putting inoculum
3. waiting 7 days
4. addition of amylacetate (for extraction of cell free broth)
5. removing cell free broth to another vessel
6. charcoal treatment for purification
7. sodium acetate for crystallization
8. filtation
9. addition ethanol
10. filtation
11. acyclase treatment+ IPTG for removing side chaine
12. addition AEH for ampicillin production

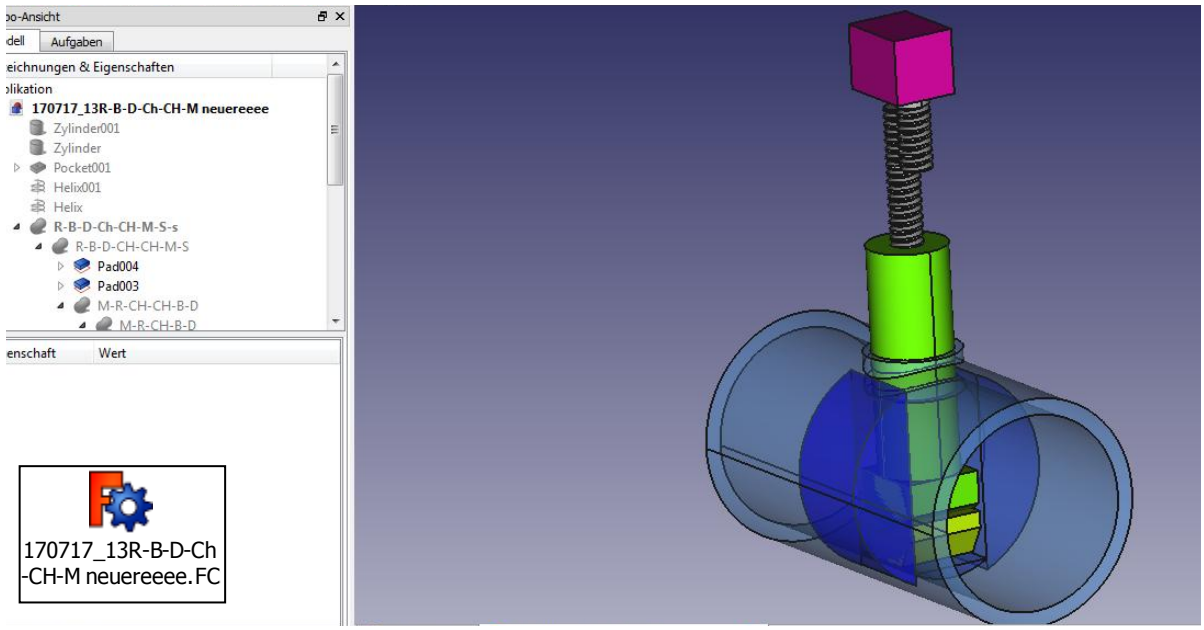

inoculum
Penicillium chy./notatum
amylacetate
charcoal treatment
sodium acetate
ethanol
Production medium:
Carbohydrate (Glucose)
Precursor(Phenyl acetic acid)
lactose (Beet molasse)
corn steep liquor
cell free broth containing penicillin
P-1 / SR-101
Seed Reactor
P-1 / SR-101
P-1 / SR-101
Seed Reactor
P-2 / UF-101
filtration
P-1 / V-101
Blending / Storage
P-2 / UF-101
filtration
P-1 / V-101
crystal drying
remaining cells
penicillin
Process sequence:
1. Sterilizing Production media
2. putting inoculum
3. waiting 7 days
4. addition of amylacetate (for extraction of cell free broth)
5. removing cell free broth to another vessel
6.charcoal treatment for purification
7.sodium acetate for crystallization
8.filtation
9.addition ethanol
10.filtation
11.acyclase treatment+ IPTG for removing side chaine
12.addition AEH for ampicillin production
AEH
acyclase treatment + IPTG
penicillin
ampicillin
P-1 / V-101
Blending / Storage
P-1 / V-101
Blending / Storage
6APA


chy./notatum
amylacetate
charcoal
treatment
sodium acetate
ethanol
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
cell free broth
containing penicillin
P-1 / SR-101
Seed Reactor
P-1 / SR-101
P-1 / SR-101
Seed Reactor
penicillin
P-2 / UF-101
filtration
P-1 / V-101
Blending / Storage
P-2 / UF-101
filtration
P-1 / V-101
Blending / Storage
crystal drying
remaining cells
acyclase
treatment + IPTG
AEH
V12
V13
V14
V15
V16
V17
V18
natural
penicillin
ampicillin
P-1 / V-101
Blending / Storage
6APA
P-1 / V-101
Blending / Storage

5 MEGBI-APP Process Control System

5.1 Automatic Valves: Conception

5.1.1 Preliminary Design of Automatic Control Valve



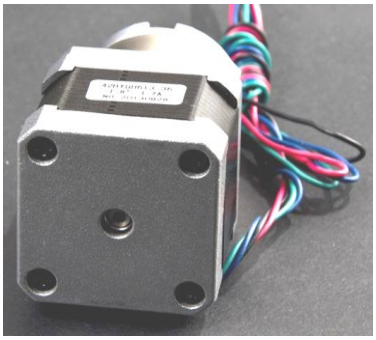
5.1.2 Alternative 1: DC Motor for automatic valves

from www.cnclablb.com: **Metal DC Geared Motor - 12V 50RPM 9kg.cm rated torque,** Price : 15.95$, Serial number : ACT0022

Description: This is a metal DC geared motor, 100% pure copper coils, high-density molecular layer, 100:1 metal reducer, small size, large torque. The maximum torque could arrive 50 kg.cm, stable and durable!

Specification: Rated voltage: 12 V, Gear reduction ratio: 100:1, D output shaft diameter: 6 mm, No-load speed: 50 RPM @ 12 v, No-load current: 0.17 A, Rated speed: 45 RPM @ 12 v, Current rating: 0.68 A, Rated torque: 9 kg.cm, Locked-rotor torque: 50 kg.cm, Locked-rotor current: 2.19 A, Power: 5W, Weight: 210 g, **Shipping List:** Metal DC Geared Motor - 12V 50RPM 50kg.cm x1

5.1.3 Alternative 2: Stepper Motor



From www.cnclablb.com

from www.cnclablb.com: Bipolar Stepper Motor with Planet Gear Box (18kg.cm), Price : 40$, Serial number : ACT0017, **!!!needs additional drive!!!**

5.1.4 Alternative3: Servo

5.1.4.1 Low Cost Servo

	from www.cnclablb.com: Metal Gear Servo TowerPro MG995 Servo - 9kg, Price : 8$ Serial number : ACT0005 **Description:** Modulation: Digital, Torque: 4.8V: 130.54 oz-in (**9.40 kg-cm**) 6.0V: 152.76 oz-in (11.00 kg-cm) Speed: 4.8V: 0.20 sec/60° 6.0V: 0.16 sec/60°, Weight: 1.94 oz (55.0 g), Dimensions:Length:1.60 in (40.7 mm), Width:0.78 in (19.7 mm), Height:1.69 in (42.9 mm)

5.1.4.2 High cost Servo

	DF15MG Tilt/Pan Kit, Price : 47.5$, Mark : DFRobot, Serial number : FIT0046 This is a 2DOF Pan and Tilt Kit assembly for horizontal surface mount. It equipped with a DF15MG servo which offers 15 kg high-torque

5.2 Actual Motorized Valve Implementation

5.2.1 Hardware and Electronics

5.2.1.1 Adopted Motor: Low Cost Servo (Alternative 3 (Low Cost Variante)

The adopted motor is the TowerPro MG995 DC Servo Motor with the following specs:

- Modulation: Digital
- Torque: 4.8V: 9.40 kg-cm 6.0V: 11.00 kg-cm
- Speed: 4.8V: 0.20 sec/60° 6.0V: 0.16 sec/60°
- Weight: 1.94 oz (55.0 g)
- Dimensions:Length:1.60 in (40.7 mm)
- Width:0.78 in (19.7 mm)
- Height:1.69 in (42.9 mm)
- LINK – CNC LAB Shop

Figure 5-1 – TowerPro MG995

The adopted motor provides the required torque to turn the ball valve.

A set of 18 servos are used with a control unit shown in 6.2.2 to allow opening and closing of 18 ball valves.

5.2.1.2 Motor Controller and Interfaces

To accommodate 18 servo motors and ensure best response the Arduino Mega 2560 was chosen for the following reasons:

- Enough PPM capable IO count to control the servos. The Arduino Mega 2560 allows control of 48 Servo motors while most of other Arduino boards allow control of only 12 servos max.
- Availability of an IO shield that makes powering and connecting all the servos much more convenient and much less time consuming.

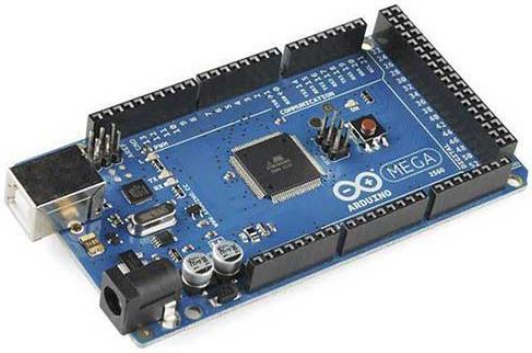


Figure 5-2 – Arduino Mega - LINK

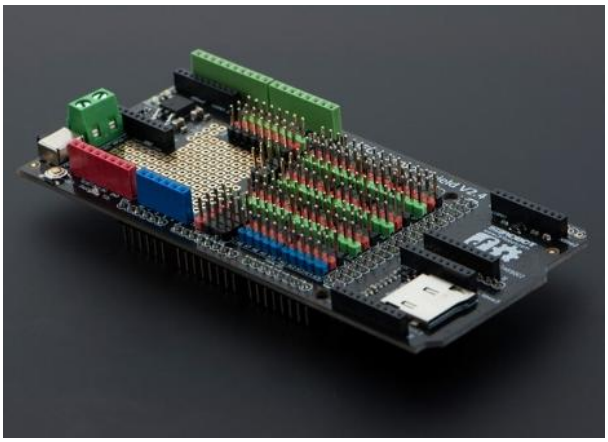


Figure 5-3 – Mega Sensor Shield - LINK

Interfacing between MEGBI python GUI and the servos can be accomplished in two ways:

a. Via Digital input signals on the Arduino Shield.
b. Via Communication through the Arduino USB port.

Digital interface mode and communication mode can be used at the same time if necessary.

The following IO map illustrates the IO allocation for the servos and the digital inputs on the Arduino Shield:

VAVLE ID	COMMAND PIN (ARDUINO INPUT)	SERVO PIN (ARDUINO OUTPUT)
1	DIO 33	DIO 14
2	DIO 34	DIO 15
3	DIO 35	DIO 16
4	DIO 36	DIO 17
5	DIO 37	DIO 18
6	DIO 38	DIO 19
7	DIO 39	DIO 20
8	DIO 40	DIO 21
9	DIO 41	DIO 22
10	DIO 42	DIO 23
11	DIO 43	DIO 24
12	DIO 44	DIO 25
13	DIO 45	DIO 26
14	DIO 46	DIO 27
15	DIO 47	DIO 28
16	DIO 48	DIO 29
17	DIO 49	DIO 30
18	DIO 50	DIO 31

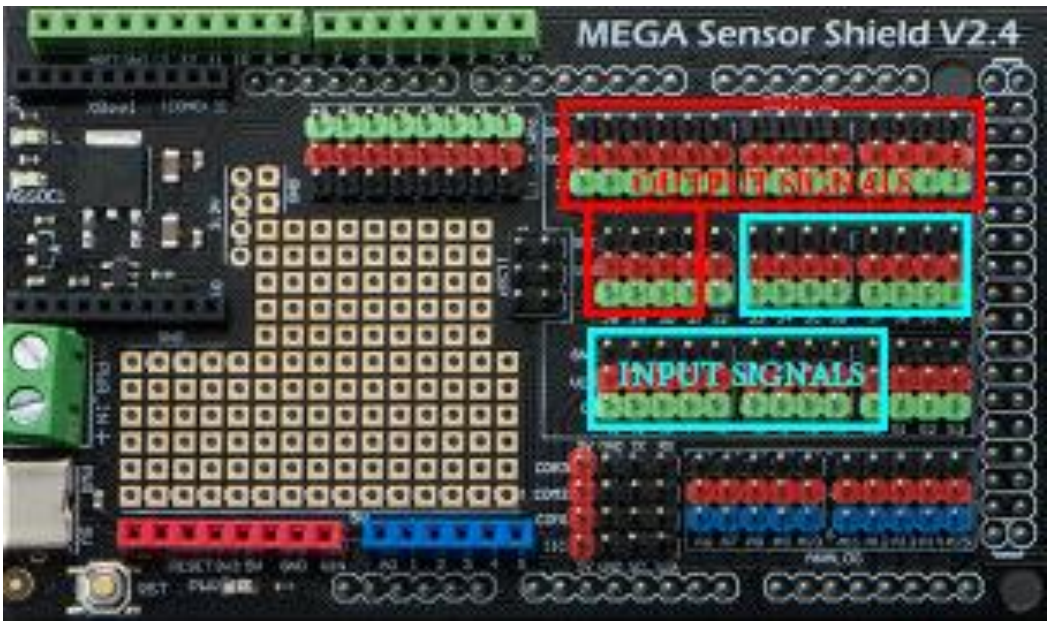


Figure 5-4 – Inputs and Outputs Allocation

The digital input mode of control allows closing and openning the valves by set or clearing the corresponding DIO respectively.

On the other hand, controlling the valves via USB communication with Arduino is implemented in an example Python code using a couple of Python classes discussed in more details in part 6.2.2.

5.2.1.3 Power Management

One of the reasons of choosing Arduino Mega IO shield was powering the motors as mentioned earlier, as 18 Servo motors can consume a hefty amount of power.

Each servo motor can consume up to 1.2 Amps at 5V at certain moments when closing or openning the valves. Thus in terms of power management the following mesures were taken:

- The IO shield allows powering the servos from a separate power connector (Green screw terminal in Fig6-4) thus isolating the limited Arduino regulator from motors consumption and ensuring microcontroller chip performance and functionality.
- Within the Arduino Firmware, precautions were taken so that the servos are only consuming power while opening or closing and for a limited time beyond that. After the time delay of a motor's activity the motor is powered down to cut its consumption to almost zero Amps.

Having mentioned the above points, selecting the motors power supply is highly related to the number of motors that are expected to be active simultaneously. For example, if the automatic mode of the plant requires that 6 motors have to be active at a certain moment; and active means is currently in the process of opening or closing; then the power supply should be a 5 VDC with at least 6 x 1.2A = 7.2 Amps.

The arduino board itself can be powered either by a USB cable connected to PC or by any standard wall adapter with voltage between 7.4V and 12V.

5.2.2 Firmware and Software

5.2.2.1 Arduino Firmware

The Arduino controller is loaded with a firmware featuring the following:

- Control of 18 Servo motors with preset positions for closed and opened valve.
- Digital Input control for all 18 valves.
- Communication protocol class for two way communication with Python GUI on PC.
- Power management for all motors.

The firmware was developed by CNC LAB. The code is developed with maintenance and scalability in mind.


ArduinoSourceCode.zip

CommandMessanger.h

```
/*
CmdMessenger - library that provides command based messaging
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
#ifndef CmdMessenger_h
#define CmdMessenger_h
#include <inttypes.h>
#if ARDUINO >= 100
#include <Arduino.h>
#else
#include <WProgram.h>
#endif
//#include "Stream.h"
extern "C"
{
// callback functions always follow the signature: void cmd(void);
typedef void(*messengerCallbackFunction) (void);
}
#define MAXCALLBACKS 50 // The maximum number of commands (default: 50)
#define MESSENGERBUFFERSIZE 64 // The length of the commandbuffer (default: 64)
#define MAXSTREAMBUFFERSIZE 512 // The length of the streambuffer (default: 64)
#define DEFAULT_TIMEOUT 5000 // Time out on unanswered messages. (default: 5s)
// Message States
enum
{
kProccesingMessage, // Message is being received, not reached command separator
kEndOfMessage, // Message is fully received, reached command separator
kProcessingArguments, // Message is received, arguments are being read parsed
};
#define white_space(c) ((c) == ' ' || (c) == '\t')
#define valid_digit(c) ((c) >= '0' && (c) <= '9')

class CmdMessenger
{
private:
// **** Private variables ***
bool startCommand; // Indicates if sending of a command is underway
```

```
uint8_t lastCommandId; // ID of last received command
uint8_t bufferIndex; // Index where to write data in buffer
uint8_t bufferLength; // Is set to MESSENGERBUFFERSIZE
uint8_t bufferLastIndex; // The last index of the buffer
char ArglastChar; // Bookkeeping of argument escape char
char CmdlastChar; // Bookkeeping of command escape char
bool pauseProcessing; // pauses processing of new commands, during sending
bool print_newlines; // Indicates if \r\n should be added after send command
char commandBuffer[MESSENGERBUFFERSIZE]; // Buffer that holds the data
char streamBuffer[MAXSTREAMBUFFERSIZE]; // Buffer that holds the data
uint8_t messageState; // Current state of message processing
bool dumped; // Indicates if last argument has been externally read
bool ArgOk; // Indicated if last fetched argument could be read
char *current; // Pointer to current buffer position
char *last; // Pointer to previous buffer position
char prevChar; // Previous char (needed for unescaping)
Stream *comms; // Serial data stream
char command_separator; // Character indicating end of command (default: ';')
char field_separator; // Character indicating end of argument (default: ',')
char escape_character; // Character indicating escaping of special chars
messengerCallbackFunction default_callback; // default callback function
messengerCallbackFunction callbackList[MAXCALLBACKS]; // list of attached callback
functions
// **** Initialize ****
void init(Stream & comms, const char fld_separator, const char cmd_separator, const char
esc_character);
void reset();
// **** Command processing ****
inline uint8_t processLine(char serialChar) __attribute__((always_inline));
inline void handleMessage() __attribute__((always_inline));
inline bool blockedTillReply(unsigned int timeout = DEFAULT_TIMEOUT, byte ackCmdId = 1)
__attribute__((always_inline));
inline bool checkForAck(byte AckCommand) __attribute__((always_inline));
// **** Command sending ****
/**
* Print variable of type T binary in binary format
*/
template < class T >
void writeBin(const T & value)
{
const byte *bytePointer = (const byte *)(const void *)&value;
for (unsigned int i = 0; i < sizeof(value); i++)
{
printEsc(*bytePointer);
bytePointer++;
}
}
// **** Command receiving ****
int findNext(char *str, char delim);
/**
* Read a variable of any type in binary format
*/
template < class T >
T readBin(char *str)
{
T value;
unescape(str);
byte *bytePointer = (byte *)(const void *)&value;
for (unsigned int i = 0; i < sizeof(value); i++)
{
*bytePointer = str[i];
bytePointer++;
}
return value;
}
template < class T >
```

```
T empty()
{
T value;
byte *bytePointer = (byte *)(const void *)&value;
for (unsigned int i = 0; i < sizeof(value); i++)
{
*bytePointer = '\0';
bytePointer++;
}
return value;
}
// **** Escaping tools ****
char *split_r(char *str, const char delim, char **nextp);
bool isEscaped(char *currChar, const char escapeChar, char *lastChar);
void printEsc(char *str);
void printEsc(char str);
public:
// ****** Public functions ******
// **** Initialization ****
CmdMessenger(Stream & comms, const char fld_separator = ',',
const char cmd_separator = ';',
const char esc_character = '/');
void printLfCr(bool addNewLine = true);
void attach(messengerCallbackFunction newFunction);
void attach(byte msgId, messengerCallbackFunction newFunction);
// **** Command processing ****
void feedinSerialData();
bool next();
bool available();
bool isArgOk();
uint8_t commandID();
// **** Command sending ****
/**
* Send a command with a single argument of any type
* Note that the argument is sent as string
*/
template < class T >
bool sendCmd(byte cmdId, T arg, bool reqAc = false, byte ackCmdId = 1,
unsigned int timeout = DEFAULT_TIMEOUT)
{
if (!startCommand) {
sendCmdStart(cmdId);
sendCmdArg(arg);
return sendCmdEnd(reqAc, ackCmdId, timeout);
}
return false;
}
/**
* Send a command with a single argument of any type
* Note that the argument is sent in binary format
*/
template < class T >
bool sendBinCmd(byte cmdId, T arg, bool reqAc = false, byte ackCmdId = 1,
unsigned int timeout = DEFAULT_TIMEOUT)
{
if (!startCommand) {
sendCmdStart(cmdId);
sendCmdBinArg(arg);
return sendCmdEnd(reqAc, ackCmdId, timeout);
}
return false;
}
bool sendCmd(byte cmdId);
bool sendCmd(byte cmdId, bool reqAc, byte ackCmdId);
// **** Command sending with multiple arguments ****
void sendCmdStart(byte cmdId);
```

```
void sendCmdEscArg(char *arg);
void sendCmdfArg(char *fmt, ...);
bool sendCmdEnd(bool reqAc = false, byte ackCmdId = 1, unsigned int timeout =
DEFAULT_TIMEOUT);
/**
* Send a single argument as string
* Note that this will only succeed if a sendCmdStart has been issued first
*/
template < class T > void sendCmdArg(T arg)
{
if (startCommand) {
comms->print(field_separator);
comms->print(arg);
}
}
/**
* Send a single argument as string with custom accuracy
* Note that this will only succeed if a sendCmdStart has been issued first
*/
template < class T > void sendCmdArg(T arg, unsigned int n)
{
if (startCommand) {
comms->print(field_separator);
comms->print(arg, n);
}
}
/**
* Send double argument in scientific format.
* This will overcome the boundary of normal d sending which is limited to abs(f) <=
MAXLONG
*/
void sendCmdSciArg(double arg, unsigned int n = 6);
/**
* Send a single argument in binary format
* Note that this will only succeed if a sendCmdStart has been issued first
*/
template < class T > void sendCmdBinArg(T arg)
{
if (startCommand) {
comms->print(field_separator);
writeBin(arg);
}
}
// **** Command receiving ****
bool readBoolArg();
int16_t readInt16Arg();
int32_t readInt32Arg();
char readCharArg();
float readFloatArg();
double readDoubleArg();
char *readStringArg();
void copyStringArg(char *string, uint8_t size);
uint8_t compareStringArg(char *string);
/**
* Read an argument of any type in binary format
*/
template < class T > T readBinArg()
{
if (next()) {
dumped = true;
return readBin < T >(current);
}
else {
return empty < T >();
}
}
// **** Escaping tools ****
```

```
void unescape(char *fromChar);
void printSci(double f, unsigned int digits);
};
#endif
```

CommandMsg.cpp

```
/*
CmdMessenger - library that provides command based messaging
Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:
The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Initial Messenger Library - Thomas Ouellet Fredericks.
CmdMessenger Version 1 - Neil Dudman.
CmdMessenger Version 2 - Dreamcat4.
CmdMessenger Version 3 - Thijs Elenbaas.
3.6 - Fixes
- Better compatibility between platforms
- Unit tests
3.5 - Fixes, speed improvements for Teensy
3.4 - Internal update
3.3 - Fixed warnings
- Some code optimization
3.2 - Small fixes and sending long argument support
3.1 - Added examples
3.0 - Bugfixes on 2.2
- Wait for acknowlegde
- Sending of common type arguments (float, int, char)
- Multi-argument commands
- Escaping of special characters
- Sending of binary data of any type (uses escaping)
*/
extern "C" {
#include <stdlib.h>
#include <stdarg.h>
}
#include <stdio.h>
#include "CmdMessenger.h"
#define _CMDMESSENGER_VERSION 3_6 // software version of this library
// **** Initialization ****
/**
* CmdMessenger constructor
*/

CmdMessenger::CmdMessenger(Stream &ccomms, const char fld_separator, const char
cmd_separator, const char esc_character)
{
init(ccomms, fld_separator, cmd_separator, esc_character);
}
/**
* Enables printing newline after a sent command
*/
void CmdMessenger::init(Stream &ccomms, const char fld_separator, const char cmd_separator,
const char esc_character)
{
default_callback = NULL;
comms = &ccomms;
```

```
print_newlines = false;
field_separator = fld_separator;
command_separator = cmd_separator;
escape_character = esc_character;
bufferLength = MESSENGERBUFFERSIZE;
bufferLastIndex = MESSENGERBUFFERSIZE - 1;
reset();
default_callback = NULL;
for (int i = 0; i < MAXCALLBACKS; i++)
callbackList[i] = NULL;
pauseProcessing = false;
}
/**
* Resets the command buffer and message state
*/
void CmdMessenger::reset()
{
bufferIndex = 0;
current = NULL;
last = NULL;
dumped = true;
}
/**
* Enables printing newline after a sent command
*/
void CmdMessenger::printLfCr(bool addNewLine)
{
print_newlines = addNewLine;
}
/**
* Attaches an default function for commands that are not explicitly attached
*/
void CmdMessenger::attach(messengerCallbackFunction newFunction)
{
default_callback = newFunction;
}
/**

* Attaches a function to a command ID
*/
void CmdMessenger::attach(byte msgId, messengerCallbackFunction newFunction)
{
if (msgId >= 0 && msgId < MAXCALLBACKS)
callbackList[msgId] = newFunction;
}
// **** Command processing ****
/**
* Feeds serial data in CmdMessenger
*/
void CmdMessenger::feedinSerialData()
{
while (!pauseProcessing && comms->available())
{
// The Stream class has a readBytes() function that reads many bytes at once. On
Teensy 2.0 and 3.0, readBytes() is optimized.
// Benchmarks about the incredible difference it makes:
http://www.pjrc.com/teensy/benchmark_usb_serial_receive.html
size_t bytesAvailable = min(comms->available(), MAXSTREAMBUFFERSIZE);
comms->readBytes(streamBuffer, bytesAvailable);
// Process the bytes in the stream buffer, and handles dispatches callbacks, if
commands are received
for (size_t byteNo = 0; byteNo < bytesAvailable; byteNo++)
{
int messageState = processLine(streamBuffer[byteNo]);
// If waiting for acknowledge command
if (messageState == kEndOfMessage)
{
handleMessage();
```

```
}
}
}
}
/**
* Processes bytes and determines message state
*/
uint8_t CmdMessenger::processLine(char serialChar)
{
messageState = kProccesingMessage;
//char serialChar = (char)serialByte;
bool escaped = isEscaped(&serialChar, escape_character, &CmdlastChar);
if ((serialChar == command_separator) && !escaped) {
commandBuffer[bufferIndex] = 0;
if (bufferIndex > 0) {
messageState = kEndOfMessage;
current = commandBuffer;
CmdlastChar = '\0';
}
reset();
}
else {
commandBuffer[bufferIndex] = serialChar;
bufferIndex++;
if (bufferIndex >= bufferLastIndex) reset();
}
return messageState;
}
/**
* Dispatches attached callbacks based on command
*/
void CmdMessenger::handleMessage()
{
lastCommandId = readInt16Arg();
// if command attached, we will call it
if (lastCommandId >= 0 && lastCommandId < MAXCALLBACKS && ArgOk &&
callbackList[lastCommandId] != NULL)
(*callbackList[lastCommandId])();
else // If command not attached, call default callback (if attached)
if (default_callback != NULL) (*default_callback)();
}
/**
* Waits for reply from sender or timeout before continuing
*/
bool CmdMessenger::blockedTillReply(unsigned int timeout, byte ackCmdId)
{
unsigned long time = millis();
unsigned long start = time;
bool receivedAck = false;
while ((time - start) < timeout && !receivedAck) {
time = millis();
receivedAck = checkForAck(ackCmdId);
}
return receivedAck;
}
/**
* Loops as long data is available to determine if acknowledge has come in
*/
bool CmdMessenger::checkForAck(byte ackCommand)
{
while (comms->available()) {
//Processes a byte and determines if an acknowlegde has come in
int messageState = processLine(comms->read());
if (messageState == kEndOfMessage) {
int id = readInt16Arg();
if (ackCommand == id && ArgOk) {
return true;
```

```
}
else {
return false;
}
}
return false;
}
return false;
}
/**
* Gets next argument. Returns true if an argument is available
*/
bool CmdMessenger::next()
{
char * temppointer = NULL;
// Currently, cmd messenger only supports 1 char for the field seperator
switch (messageState) {
case kProccesingMessage:
return false;
case kEndOfMessage:
temppointer = commandBuffer;
messageState = kProcessingArguments;
default:
if (dumped)
current = split_r(temppointer, field_separator, &last);
if (current != NULL) {
dumped = true;
return true;
}
}
return false;
}
/**
* Returns if an argument is available. Alias for next()
*/
bool CmdMessenger::available()
{
return next();
}
/**
* Returns if the latest argument is well formed.
*/
bool CmdMessenger::isArgOk()
{
return ArgOk;
}
/**
* Returns the commandID of the current command
*/
uint8_t CmdMessenger::commandID()
{
return lastCommandId;
}
// **** Command sending ****
/**
* Send start of command. This makes it easy to send multiple arguments per command
*/
void CmdMessenger::sendCmdStart(byte cmdId)
{
if (!startCommand) {
startCommand = true;
pauseProcessing = true;
comms->print(cmdId);
}
}
/**
* Send an escaped command argument
```

```
*/
void CmdMessenger::sendCmdEscArg(char* arg)
{
if (startCommand) {
comms->print(field_separator);
printEsc(arg);
}
}
/**
* Send formatted argument.
* Note that floating points are not supported and resulting string is limited to 128 chars
*/
void CmdMessenger::sendCmdfArg(char *fmt, ...)
{
const int maxMessageSize = 128;
if (startCommand) {
char msg[maxMessageSize];
va_list args;
va_start(args, fmt);
vsnprintf(msg, maxMessageSize, fmt, args);
va_end(args);
comms->print(field_separator);
comms->print(msg);
}
}
/**
* Send double argument in scientific format.
* This will overcome the boundary of normal float sending which is limited to abs(f) <=
MAXLONG
*/
void CmdMessenger::sendCmdSciArg(double arg, unsigned int n)
{
if (startCommand)
{
comms->print(field_separator);
printSci(arg, n);
}
}
/**
* Send end of command
*/
bool CmdMessenger::sendCmdEnd(bool reqAc, byte ackCmdId, unsigned int timeout)
{
bool ackReply = false;
if (startCommand) {
comms->print(command_separator);
if (print_newlines)
comms->println(); // should append BOTH \r\n
if (reqAc) {
ackReply = blockedTillReply(timeout, ackCmdId);
}
}
pauseProcessing = false;
startCommand = false;
return ackReply;
}
/**
* Send a command without arguments, with acknowledge
*/
bool CmdMessenger::sendCmd(byte cmdId, bool reqAc, byte ackCmdId)
{
if (!startCommand) {
sendCmdStart(cmdId);
return sendCmdEnd(reqAc, ackCmdId, DEFAULT_TIMEOUT);
}
return false;
}
```

```
/**
* Send a command without arguments, without acknowledge
*/
bool CmdMessenger::sendCmd(byte cmdId)
{
if (!startCommand) {
sendCmdStart(cmdId);
return sendCmdEnd(false, 1, DEFAULT_TIMEOUT);
}
return false;
}
// **** Command receiving ****
/**
* Find next argument in command
*/
int CmdMessenger::findNext(char *str, char delim)
{
int pos = 0;
bool escaped = false;
bool EOL = false;
ArglastChar = '\0';
while (true) {
escaped = isEscaped(str, escape_character, &ArglastChar);
EOL = (*str == '\0' && !escaped);
if (EOL) {
return pos;
}
if (*str == field_separator && !escaped) {
return pos;
}
else {
str++;
pos++;
}
}
return pos;
}
/**
* Read the next argument as int
*/
int16_t CmdMessenger::readInt16Arg()
{
if (next()) {
dumped = true;
ArgOk = true;
return atoi(current);
}
ArgOk = false;
return 0;
}
/**
* Read the next argument as int
*/
int32_t CmdMessenger::readInt32Arg()
{
if (next()) {
dumped = true;
ArgOk = true;
return atol(current);
}
ArgOk = false;
return 0L;
}
/**
* Read the next argument as bool
*/
bool CmdMessenger::readBoolArg()
```

```
{
return (readInt16Arg() != 0) ? true : false;
}
/**
* Read the next argument as char
*/
char CmdMessenger::readCharArg()
{
if (next()) {
dumped = true;
ArgOk = true;
return current[0];
}
ArgOk = false;
return 0;
}
/**
* Read the next argument as float
*/
float CmdMessenger::readFloatArg()
{
if (next()) {
dumped = true;
ArgOk = true;
//return atof(current);
return strtod(current, NULL);
}
ArgOk = false;
return 0;
}
/**
* Read the next argument as double
*/
double CmdMessenger::readDoubleArg()
{
if (next()) {
dumped = true;
ArgOk = true;
return strtod(current, NULL);
}
ArgOk = false;
return 0;
}
/**
* Read next argument as string.
* Note that the String is valid until the current command is replaced
*/
char* CmdMessenger::readStringArg()
{
if (next()) {
dumped = true;
ArgOk = true;
return current;
}
ArgOk = false;
return '\0';
}
/**
* Return next argument as a new string
* Note that this is useful if the string needs to be persisted
*/
void CmdMessenger::copyStringArg(char *string, uint8_t size)
{
if (next()) {
dumped = true;
ArgOk = true;
strlcpy(string, current, size);
```

```
}
else {
ArgOk = false;
if (size) string[0] = '\0';
}
}
/**
* Compare the next argument with a string
*/
uint8_t CmdMessenger::compareStringArg(char *string)
{
if (next()) {
if (strcmp(string, current) == 0) {
dumped = true;
ArgOk = true;
return 1;
}
else {
ArgOk = false;
return 0;
}
}
return 0;
}
// **** Escaping tools ****
/**
* Unescapes a string
* Note that this is done inline
*/
void CmdMessenger::unescape(char *fromChar)
{
// Move unescaped characters right
char *toChar = fromChar;
while (*fromChar != '\0') {
if (*fromChar == escape_character) {
fromChar++;
}
*toChar++ = *fromChar++;
}
// Pad string with \0 if string was shortened
for (; toChar < fromChar; toChar++) {
*toChar = '\0';
}
}
/**
* Split string in different tokens, based on delimiter
* Note that this is basically strtok_r, but with support for an escape character
*/
char* CmdMessenger::split_r(char *str, const char delim, char **nextp)
{
char *ret;
// if input null, this is not the first call, use the nextp pointer instead
if (str == NULL) {
str = *nextp;
}
// Strip leading delimiters
while (findNext(str, delim) == 0  && *str) {
str++;
}
// If this is a \0 char, return null
if (*str == '\0') {
return NULL;
}
// Set start of return pointer to this position
ret = str;
// Find next delimiter
str += findNext(str, delim);
```

```
// and exchange this for a a \0 char. This will terminate the char
if (*str) {
*str++ = '\0';
}
// Set the next pointer to this char
*nextp = str;
// return current pointer
return ret;
}
/**
* Indicates if the current character is escaped
*/
bool CmdMessenger::isEscaped(char *currChar, const char escapeChar, char *lastChar)
{
bool escaped;
escaped = (*lastChar == escapeChar);
*lastChar = *currChar;
// special case: the escape char has been escaped:
if (*lastChar == escape_character && escaped) {
*lastChar = '\0';
}
return escaped;
}
/**
* Escape and print a string
*/
void CmdMessenger::printEsc(char *str)
{
while (*str != '\0') {
printEsc(*str++);
}
}
/**
* Escape and print a character
*/
void CmdMessenger::printEsc(char str)
{
if (str == field_separator || str == command_separator || str == escape_character || str
== '\0') {
comms->print(escape_character);
}
comms->print(str);
}
/**
* Print float and double in scientific format
*/
void CmdMessenger::printSci(double f, unsigned int digits)
{
// handle sign
if (f < 0.0)
{
Serial.print('-');
f = -f;
}
// handle infinite values
if (isinf(f))
{
Serial.print("INF");
return;
}
// handle Not a Number
if (isnan(f))
{
Serial.print("NaN");
return;
}
// max digits
if (digits > 6) digits = 6;
```

```
long multiplier = pow(10, digits); // fix int => long
int exponent;
if (abs(f) < 10.0) {
exponent = 0;
}
else {
exponent = int(log10(f));
}
float g = f / pow(10, exponent);
if ((g < 1.0) && (g != 0.0))
{
g *= 10;
exponent--;
}
long whole = long(g); // single digit
long part = long((g - whole)*multiplier + 0.5); // # digits
// Check for rounding above .99:
if (part == 100) {
whole++;
part = 0;
}
char format[16];
sprintf(format, "%%ld.%%0%dldE%%+d", digits);
char output[16];
sprintf(output, format, whole, part, exponent);
comms->print(output);
}
```

ValvesControl.ino

```
#include <Servo.h>
#include "CmdMessenger.h"
//#include <MemoryFree.h>
//To control a valve use one of the following option:
// 1- Set the corresponding Valve Input Signal
// Input Pin: 33 -> 50
// Valve index: 0 -> 18
// 2- Send a serial command with the following syntax:
// v<x><y> where
// x is a character {0-9,:,;,<,?,@,A} corresponding the valve index {0-17} respectively
// y is a character {0,1} corresponding to OPEN and CLOSE respectively
//The servos are to be connected as follows:
// Output Pin: 14 -> 31
// Valve index: 0 -> 18
#define OPEN 138
#define CLOSE 35
enum {
cmd_connect,
rep_connected,
cmd_open_valve,
cmd_close_valve,
rep_valve_state,
rep_error,
};
const int BAUD_RATE = 9600;
CmdMessenger c = CmdMessenger(Serial,',',';','/');
/*
char servoCharV[18] ={'0', '1', '2', '3', '4',
'5', '6', '7', '8', '9',
':', ';', '<', '=', '>',
'?', '@', 'A'};
*/
int signalsPins[18] ={33, 34, 35, 36, 37,
38, 39, 40, 41, 42,
43, 44, 45, 46, 47,
48, 49, 50};
int servosPins[18] = {14, 15, 16, 17, 18,
```

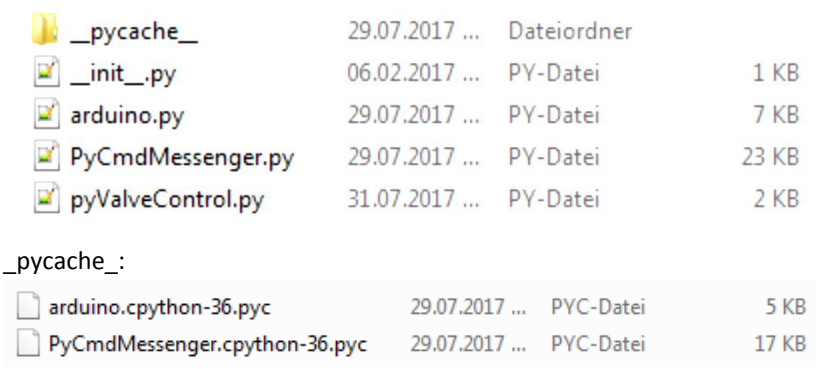
```
19, 20, 21, 22, 23,
24, 25, 26, 27, 28,
29, 30, 31};
bool virtualSignals[18];
bool preVirtualSignals[18];
bool prevInputSignals[18];
bool servoStates[18];
bool prevServoStates[18];
Servo servos[18];
long servosTimers[18];
long detachInterval = 3000;
bool anyAttached = false;
void setup()

{
pinMode(13,OUTPUT);
Serial.begin(BAUD_RATE);
attach_callbacks();
//Serial.println("Initializing Valves");
InitValves();
digitalWrite(13,HIGH);
//Serial.println("Initialization Complete");
}
//long fmpm = 0;
void loop()
{
c.feedinSerialData();
//StateMachine();
UpdateValves();
/*if(millis() - fmpm >= 1000)
{
Serial.println(freeMemory());
fmpm = millis();
}*/
}
/* callback */
void on_connect(void)
{
c.sendCmd(rep_connected,"OK");
}
/* callback */
void on_open_valve(void)
{
int value1 = c.readBinArg<int>();
if(value1 >= 0 && value1 < 18)
{
virtualSignals[value1] = true;
c.sendCmdStart (rep_valve_state);
c.sendCmdBinArg<int16_t>((int16_t)value1);
c.sendCmdBinArg<int16_t>((int16_t)1);
c.sendCmdEnd ();
//c.sendBinCmd(rep_valve_state,value1,1);
}
else
c.sendBinCmd(rep_error,"Invalid Valve Index");
}
/* callback */
void on_close_valve(void)
{
int value1 = c.readBinArg<int>();
if(value1 >= 0 && value1 < 18)
{
virtualSignals[value1] = false;
c.sendCmdStart (rep_valve_state);
c.sendCmdBinArg<int16_t>((int16_t)value1);
c.sendCmdBinArg<int16_t>((int16_t)0);

c.sendCmdEnd ();
```

```
//c.sendBinCmd(rep_valve_state,value1,0);
}
else
c.sendBinCmd(rep_error,"Invalid Valve Index");
}
/* callback */
void on_unknown_command(void)
{
c.sendCmd(rep_error,"Unknown Command");
}
/* Attach callbacks for CmdMessenger commands */
void attach_callbacks(void)
{
c.attach(cmd_connect,on_connect);
c.attach(cmd_open_valve,on_open_valve);
c.attach(cmd_close_valve,on_close_valve);
c.attach(on_unknown_command);
}
/*int machineState = 0;
int rxVIdx = -1;
void StateMachine()
{
char c;
if(Serial.available())
{
c = Serial.read();
//Serial.print("Rx: ");
//Serial.print(char(c));
//Serial.print(" - S ");
//Serial.print(machineState);
switch(machineState)
{
case 0:
if(c == 'v') machineState++;
else if(c == 'C') Serial.println("OK");
break;
case 1:
rxVIdx = int(c) - 0x30;
//Serial.print(" - IDX ");
//Serial.print(rxVIdx);
machineState++;
break;
case 2:
if(rxVIdx >= 0 && rxVIdx < 18)
{
if(c == '1') virtualSignals[rxVIdx] = true;
else if(c == '0') virtualSignals[rxVIdx] = false;
//Serial.print(" - OC ");
//Serial.print(virtualSignals[rxVIdx]? "Open":"Close");
}
machineState = 0;
rxVIdx = -1;
break;

default:
machineState = 0;
rxVIdx = -1;
break;
}
//Serial.print(" - NS ");
//Serial.println(machineState);
}
}*/
void UpdateValves()
{
for(int i = 0; i < 18; i++)
{
bool doMove = true;
bool state = !digitalRead(signalsPins[i]);
//Serial.print(!state? "1":"0");
```

```
//Serial.print("-");
if(state != prevInputSignals[i])
{
prevInputSignals[i] = state;
}
else if((preVirtualSignals[i] != virtualSignals[i]))
{
state = virtualSignals[i];
preVirtualSignals[i] = virtualSignals[i];
}
else
doMove = false;
//state = state || virtualSignals[i];
if(doMove)
ControlValve(i, state? OPEN:CLOSE, false);
//Serial.print(servoStates[i]? "1":"0");
//Serial.print("-");
}
//Serial.println();
DetachServos();
}
void InitValves()
{
bool ledState = false;
for(int i = 0; i < 18; i++)
{
pinMode(signalsPins[i], INPUT_PULLUP);
virtualSignals[i] = false;
preVirtualSignals[i] = false;
servoStates[i] = false;
prevServoStates[i] = false;
ControlValve(i, CLOSE, true);
//delay(1000);
delay(100);
DetachServos();
prevInputSignals[i] = !digitalRead(signalsPins[i]);
digitalWrite(13,ledState);
ledState = !ledState;
}

DetachServos();
}
void OpenValve(int idx)
{
ControlValve(idx, OPEN, false);
}
void CloseValve(int idx)
{
ControlValve(idx, CLOSE, false);
}
void ControlValve(int idx, int state, bool force)
{
servoStates[idx] = (state == OPEN);
virtualSignals[idx] = servoStates[idx];
if((servoStates[idx] != prevServoStates[idx]) || force)
{
//Serial.print((state == OPEN)? "Open":"Close");
//Serial.print(" Servo "); Serial.println(idx);
prevServoStates[idx] = servoStates[idx];
if(!servos[idx].attached())
servos[idx].attach(servosPins[idx]);
servos[idx].write(state);
servosTimers[idx] = millis();
anyAttached = true;
}
}
void DetachServos()
```

```
{
if(!anyAttached) return;
bool ledState = false;
bool tempAnyAttach = false;
for(int i = 0; i < 18; i++)
{
bool isat = servos[i].attached();
//Serial.print(isat? "1":"0"); Serial.print("-");
if(isat)
{
if(millis() - servosTimers[i] >= detachInterval)
servos[i].detach();
else
tempAnyAttach = true;
}
digitalWrite(13,ledState);
ledState = !ledState;
}
//Serial.println();
anyAttached = tempAnyAttach;
}
```

5.2.2.2 Python Software




PythonCode.zip

Two Python classes are available to allow two communication with Arduino:

- "arduino.py" Class defines and Arduino object with all the communication hardware settings and buffers encapsulated to send and receive general binary data. [Ref [6]] (Harms)
- "PyCmdMessenger.py" Class encapsulates a communication protocol that allows developer to define custom commands and replies and the class instance can manage and parse all communication with Arduino. . [Ref [6]] (Harms)

An additional Python code file is also included:

"pyValveControl.py" This code illustrates how to use the above mentioned classes to define the requires commands and replies that are compatible with the Arduino firmware and shows how to control the valves using the USB communication mode. [Developed by CNC LAB]

5.2.3 Retrofit,3D Models and 3D Prints

Retrofitting the ball valve with Servo motor was achieved by designing a functional mechanism that ensures the following:

- Fixating the Motor body to the Valve body to prevent motor body from rotating.
- Coupling the motor shaft with the valve shaft while improving or at least not hindering the motor torque.
- Minimize the scale factor of the mechanism.

The following design was modeled, 3D printed and tested during 3 iterations. Tweaked and optimized with each iteration.

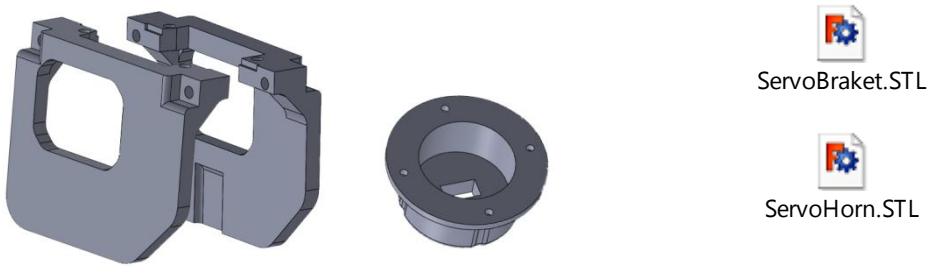


Figure 5-5 – Retrofit 3D Model

The 3D model was printed and test as shown in the following pictures:

Figure 5-6 - Out Of Printer

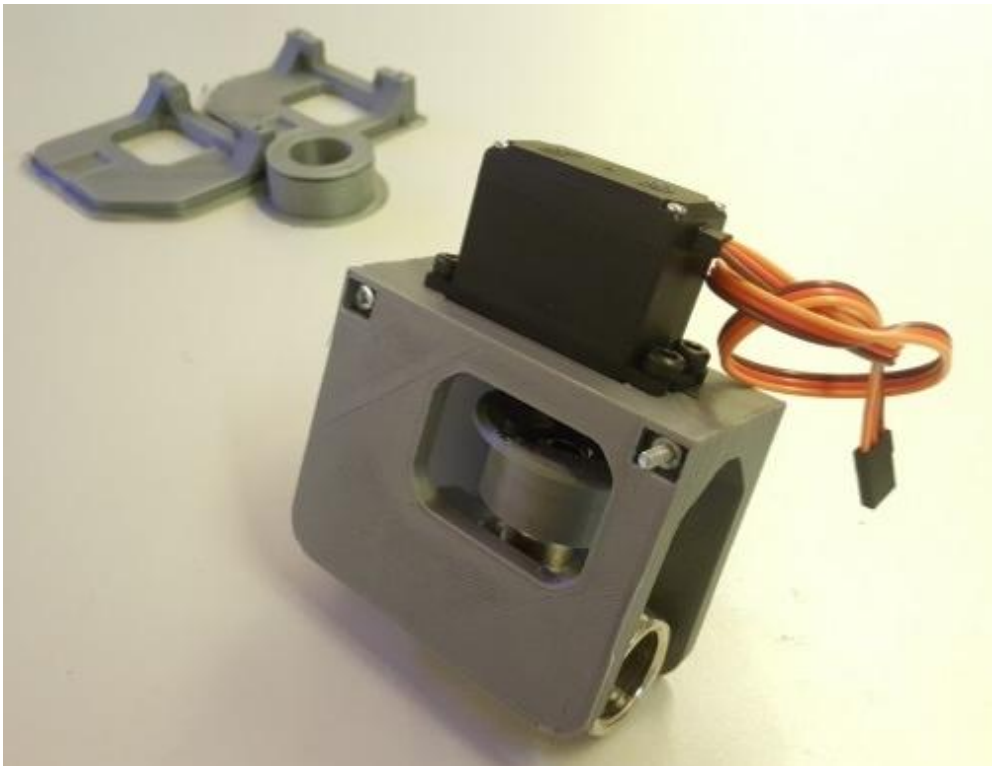


Figure 5-7 - Valve Assembly

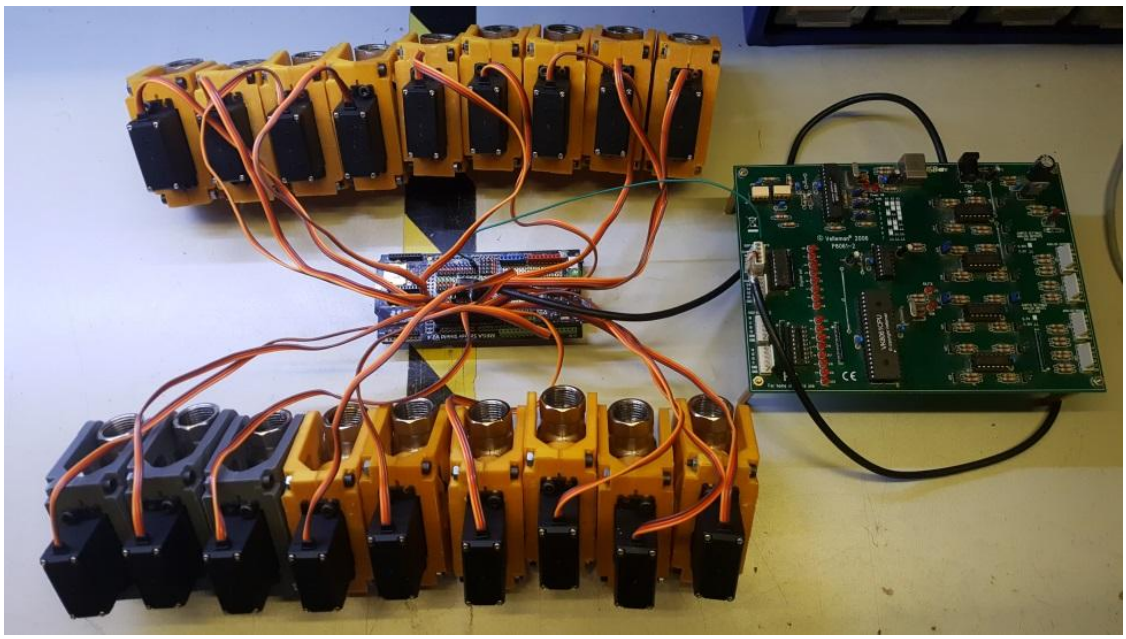
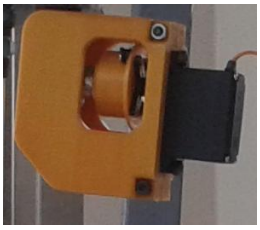


Figure 5-8 - Complete System

5.3 Integration

5.3.1 Costs

	#	Cost/#	Total
 Valve	18	$60	$1.080
	36	$2	$72
	36	$1	$36

Stand 29.10.17: Noch offen zur Beendigung des Teststandes: Anschlüsse für 18 Valves

5.3.2 Piping

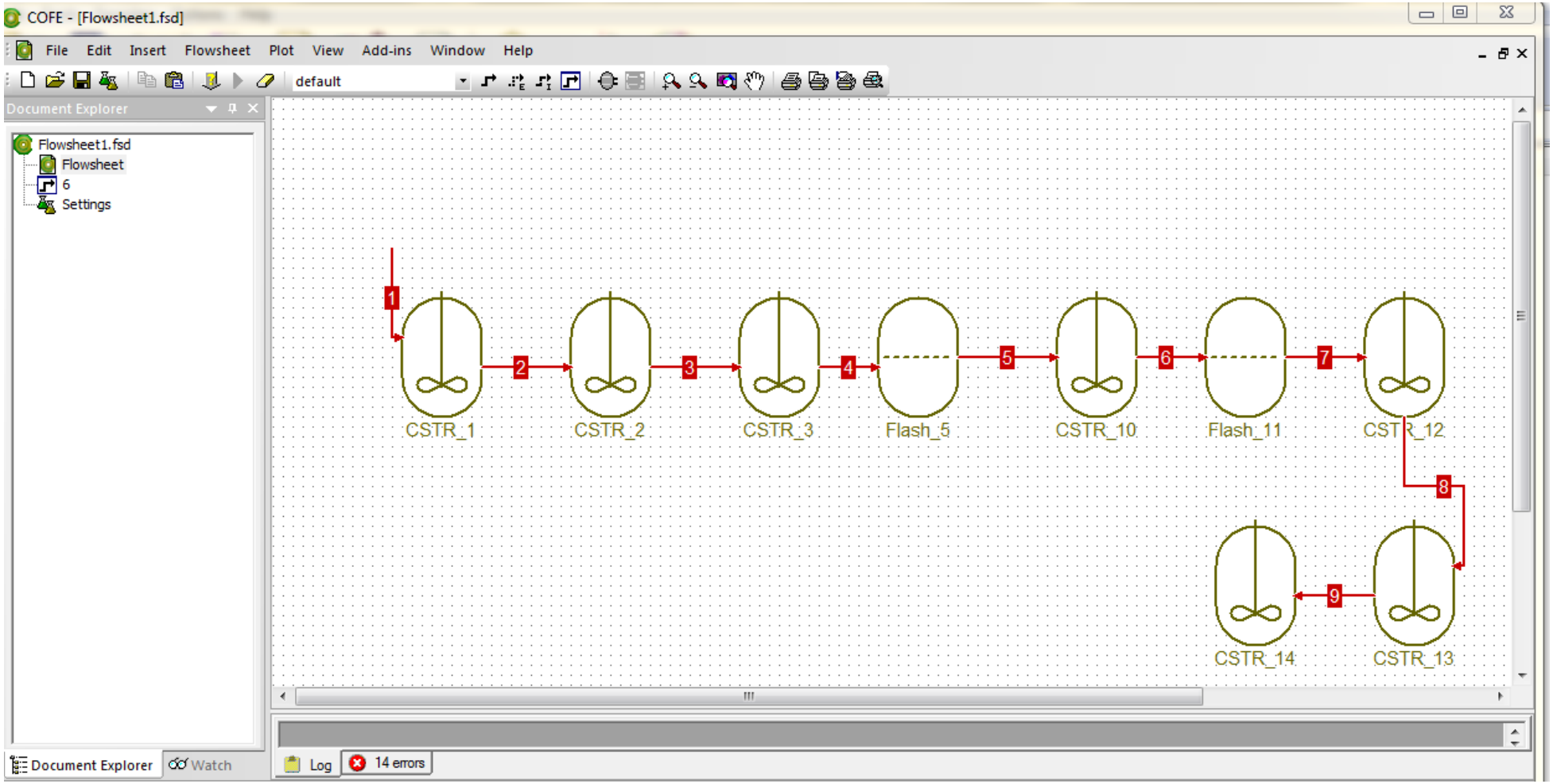
To be finished by Fatima Antar

5.3.3 Connecting to automation system

Possibility: Portating GUI to Raspberry

6 Chemical Process Simulation of MEGBI-APP

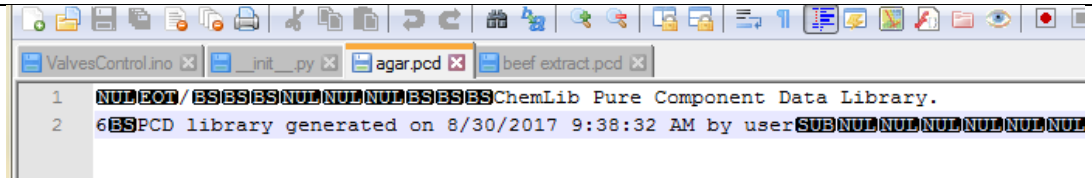
6.1 Flowsheet



6.2 Compounds (pcd files)

all	pcd-Files.zip
Agar Chloroform Corn steep liquor CuSO4 . 5H2O FeSO4 . 7H2O KCl MgSO4 . 7H2O NaNO3 Peptone Phosphate buffer Sucrose Yeast extract ZnSO4 . 7H2O	corn steep liquor.log corn steep liquor.pcd CuSO4.5H2O.log CuSO4.5H2O.pcd FeSO4.7H2O.log FeSO4.7H2O.pcd glucose.log glucose.pcd KCl.log KCl.pcd metal solution.log metal solution.pcd MgSO4·7H2O .log MgSO4·7H2O .pcd NaNO3.log NaNO3.pcd peptone.log peptone.pcd phosphate buffer.log phosphate buffer.pcd sucrose.log sucrose.pcd yeastextract.log yeastextract.pcd ZnSO4·7H2O.log ZnSO4·7H2O.pcd agar.log agar.pcd amylacetate.log amylacetate.pcd beef extract.log beef extract.pcd chloroform.log chloroform.pcd

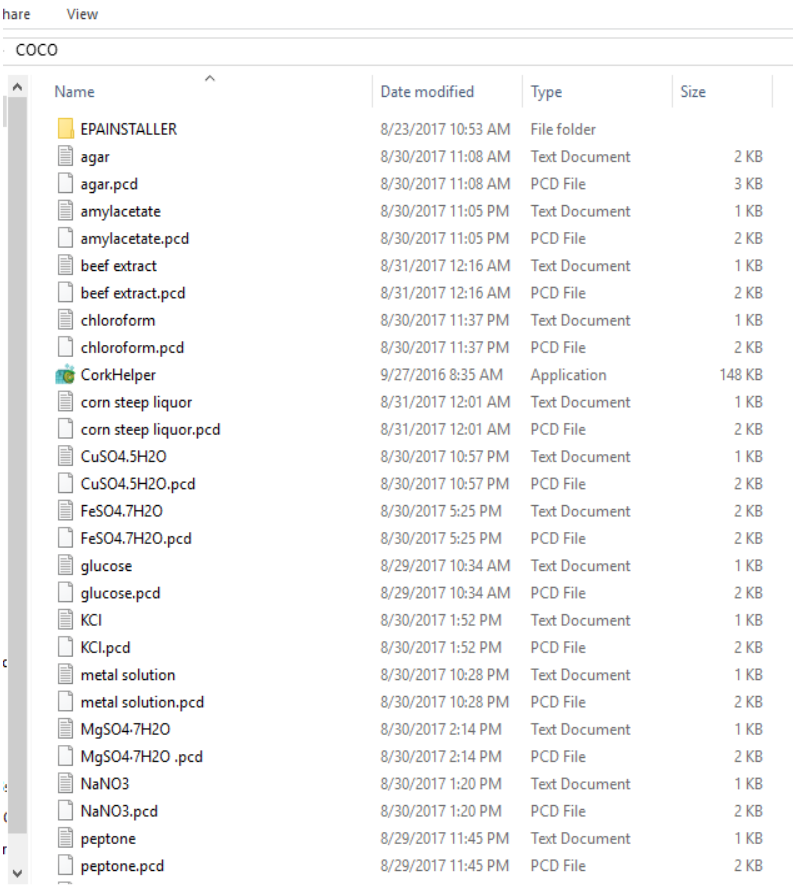
6.2.1 Example files

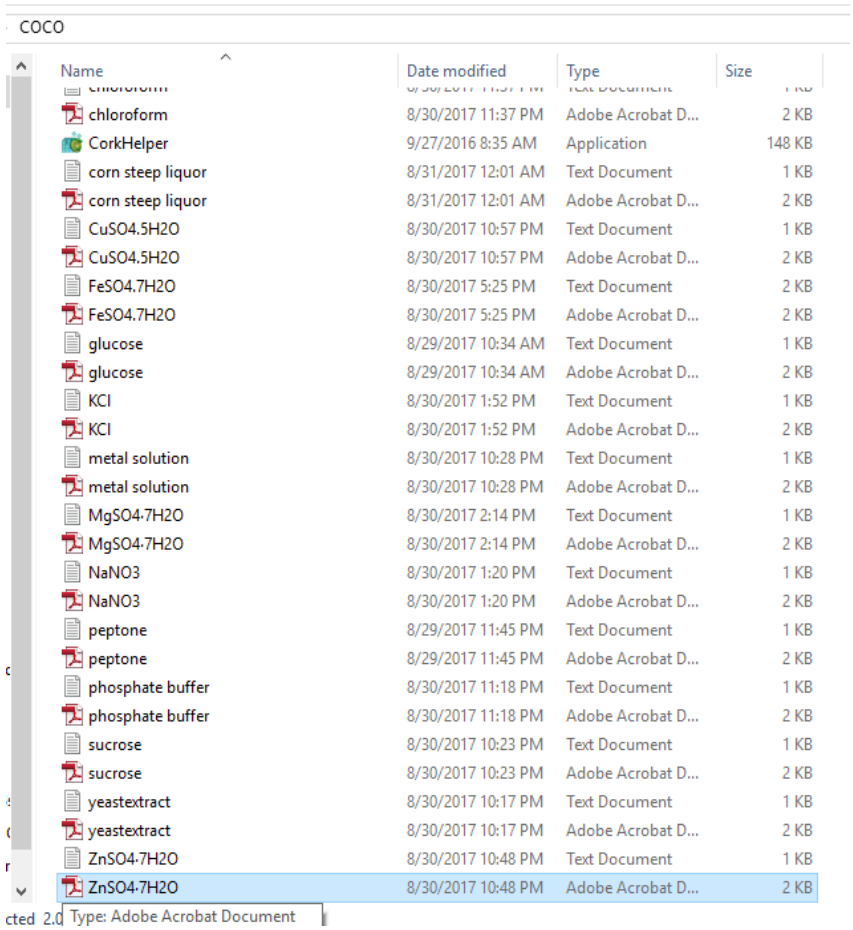
agar.log	agar.log	Added component 13097 agar agar; LIX=13097; CAS number[70]; old=; new=9002-18-0; chk=126179476; on 8/30/2017 9:45:08 AM by user agar; CAS=9002-18-0; Molecular weight[13]; old=*; new=336.337; chk=146441160; on 8/30/2017 9:51:53 AM by user agar; CAS=9002-18-0; Structure[3]; old=; new=C14H24O9; chk=177770139; on 8/30/2017 10:08:35 AM by user agar; CAS=9002-18-0; Molecular weight[13]; old=336.337; new=336.3382; chk=180295062; on 8/30/2017 10:08:39 AM by user agar; CAS=9002-18-0; Family[4]; old=0; new=73; chk=185080742; *->other polyfunctional organics on 8/30/2017 10:28:49 AM by user Added component 19599 NaNO3 NaNO3; LIX=19599; CAS number[70]; old=; new=7631-99-4; chk=133951884; on 8/30/2017 10:41:34 AM by user NaNO3; CAS=7631-99-4; Molecular weight[13]; old=*; new= 84.99; chk=180189686; on 8/30/2017 10:42:11 AM by user NaNO3; CAS=7631-99-4; Structure[3]; old=; new=NaNO3; chk=205267212; on 8/30/2017 11:04:42 AM by user NaNO3; CAS=7631-99-4; Structure[3]; old=NaNO3; new=NNaO3; chk=205267193; on 8/30/2017 11:05:16 AM by user NaNO3; CAS=7631-99-4; Molecular weight[13]; old=84.99; new=84.99467; chk=195302116; on 8/30/2017 11:05:18 AM by user NaNO3; CAS=7631-99-4; Molecular weight[13]; old=84.99467; new=84.994; chk=206263521; on 8/30/2017 11:05:34 AM by user NaNO3; CAS=7631-99-4; Family[4]; old=0; new=80; chk=211514386; *->sodium salts on 8/30/2017 11:08:35 AM by user
Agar.pcd	agar.pcd	

7 Materials for MEGBI-APP

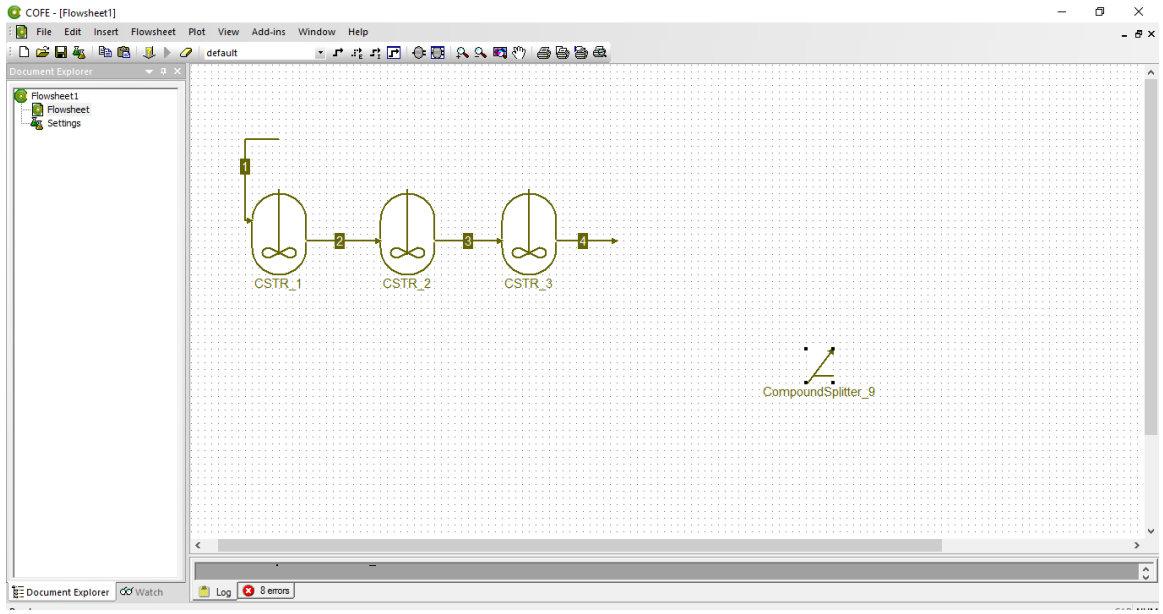
7.1 Chemicals from Sigma Aldrich

7.1.1 new compounds on coco



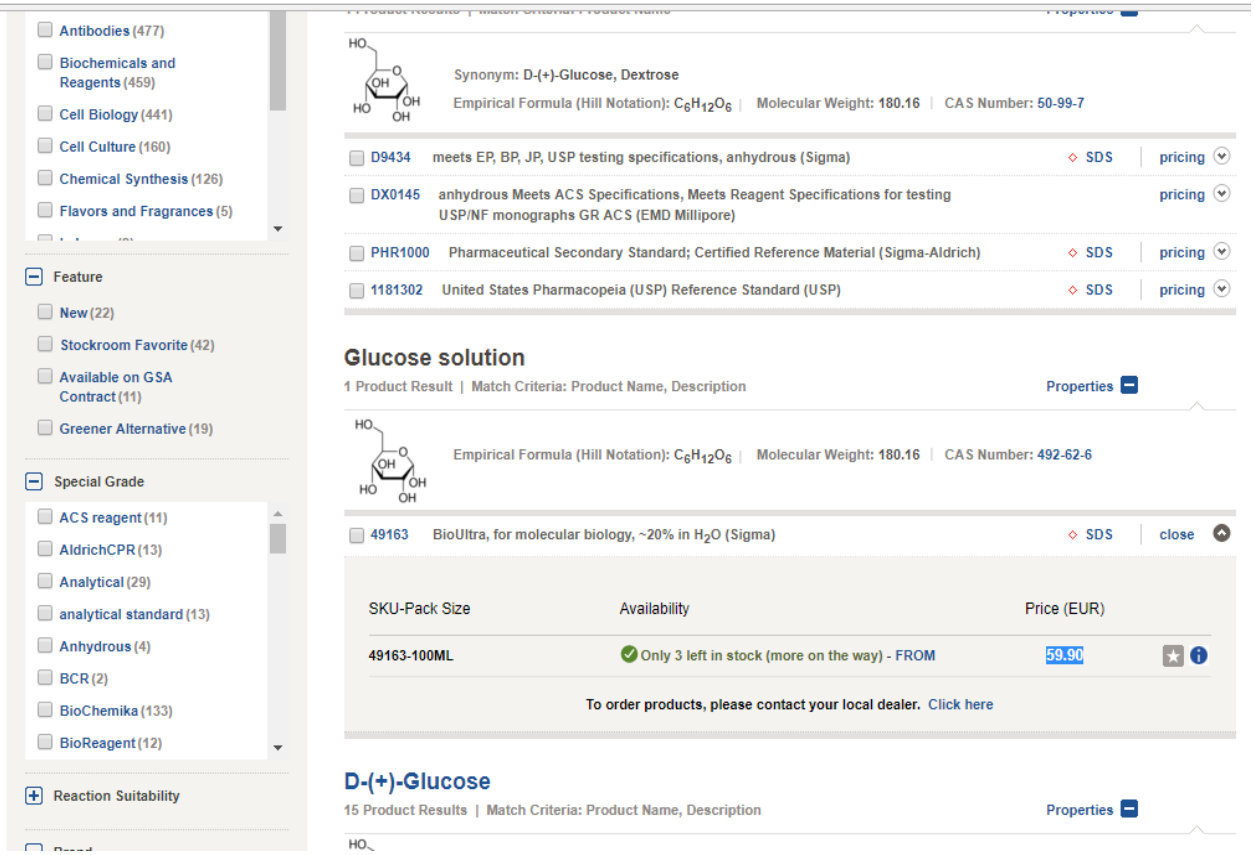


Problem: I can't use these compounds in coco (cofe 64)

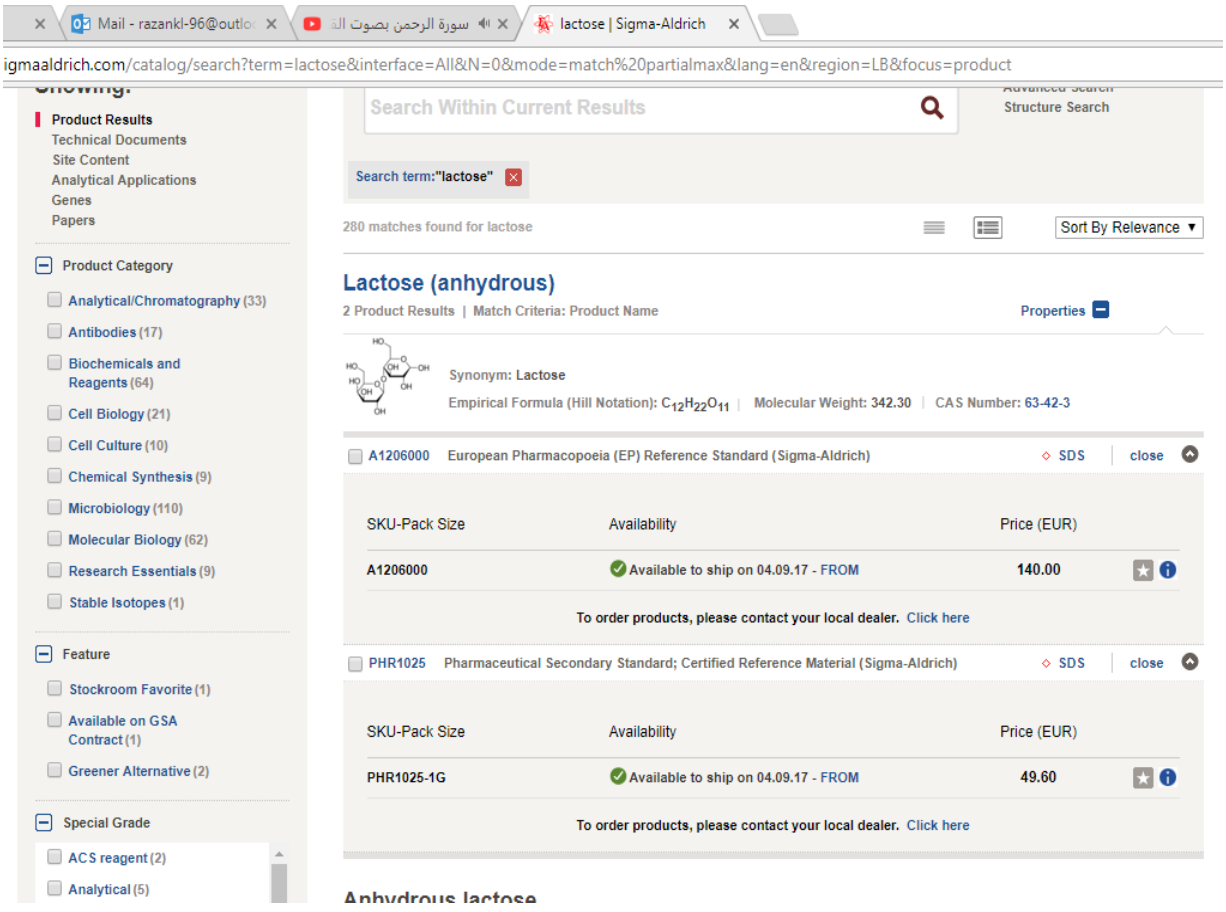


7.1.2 Glucose:

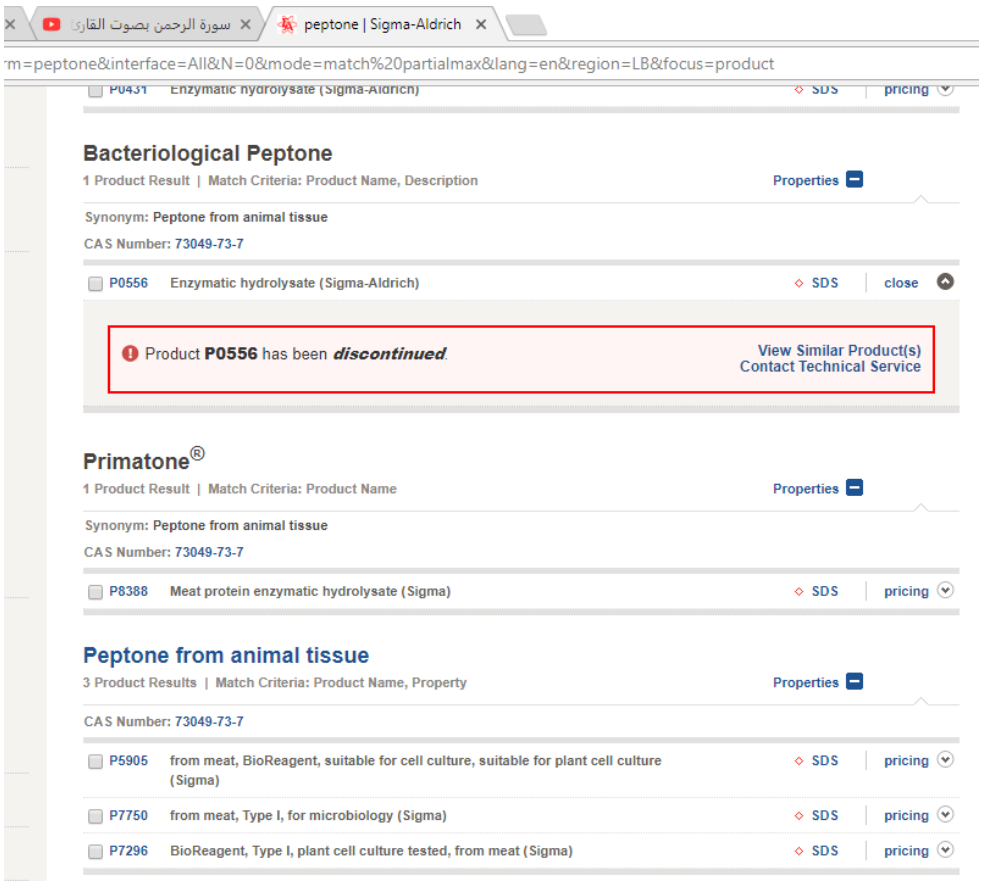
.sigmaaldrich.com/catalog/search?term=glucose&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product



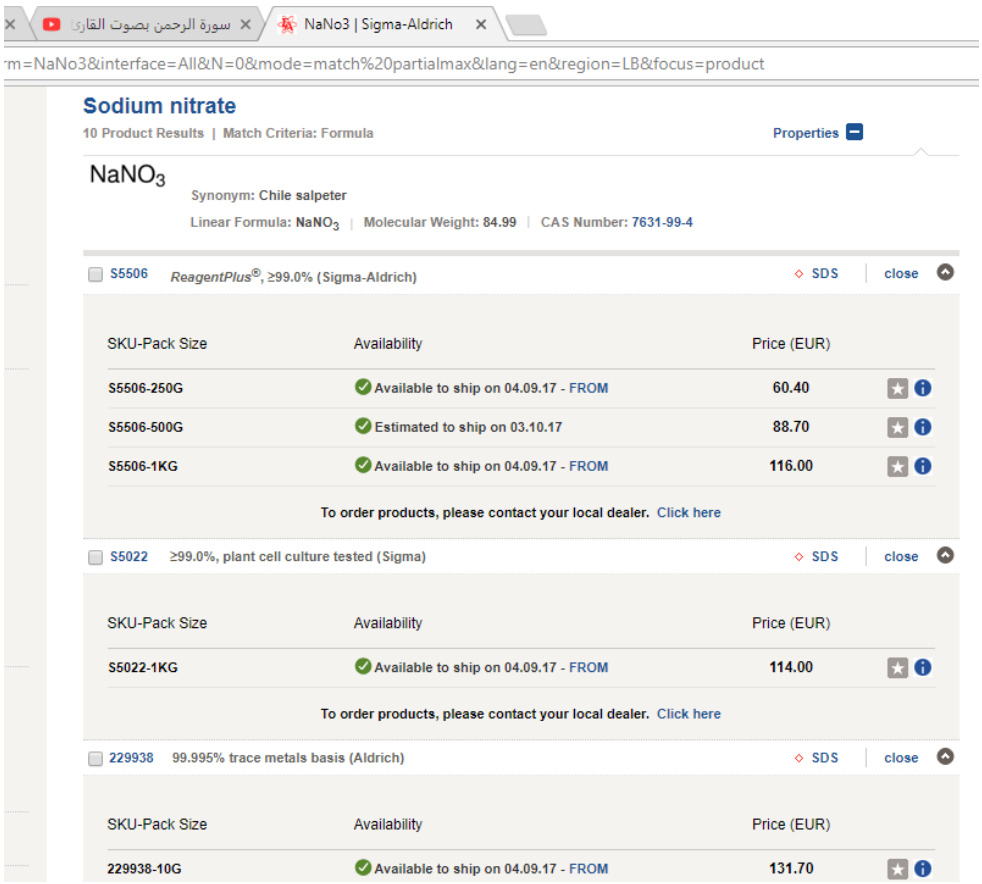
7.1.3 Lactose:

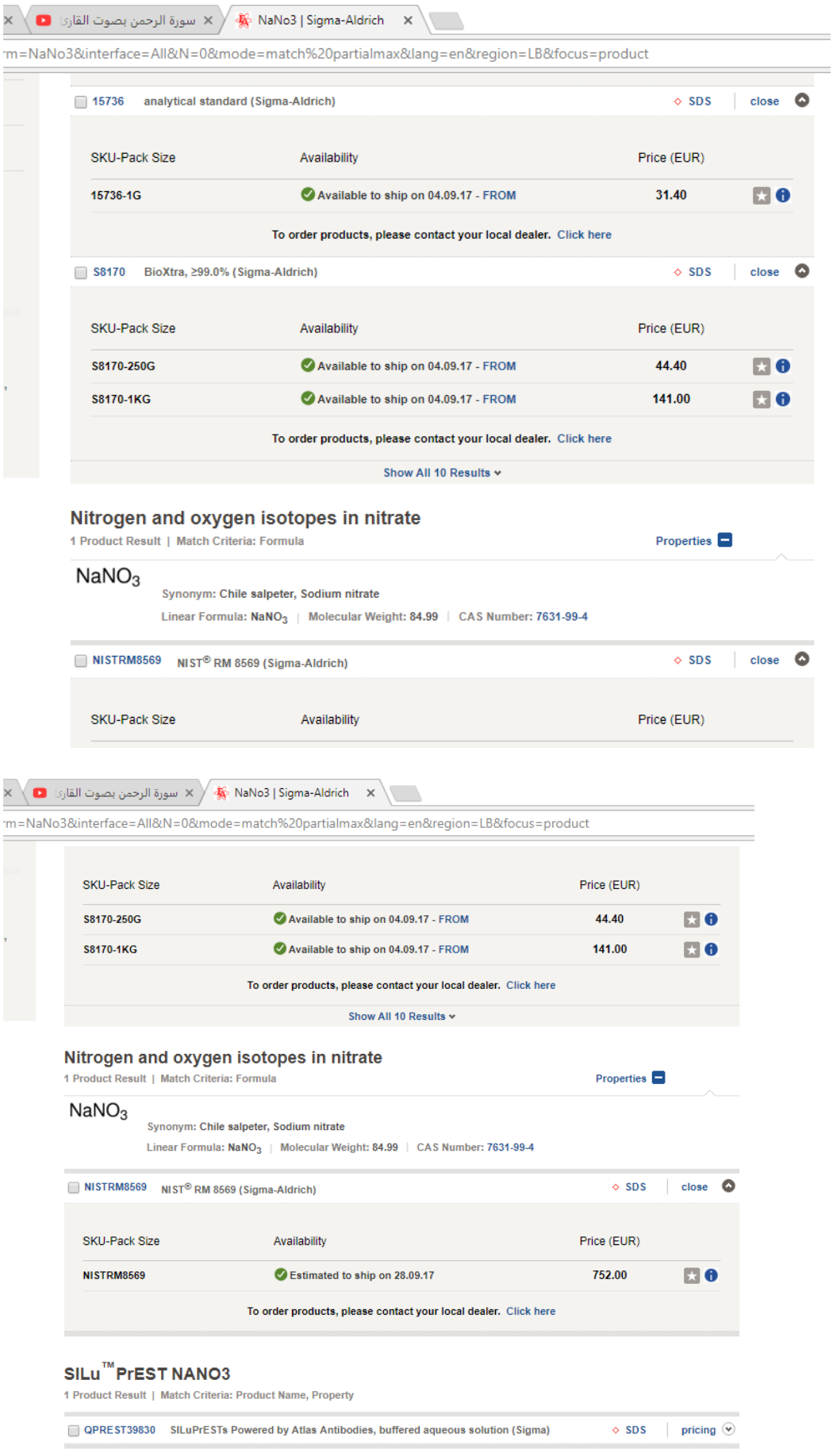


7.1.4 Peptone:

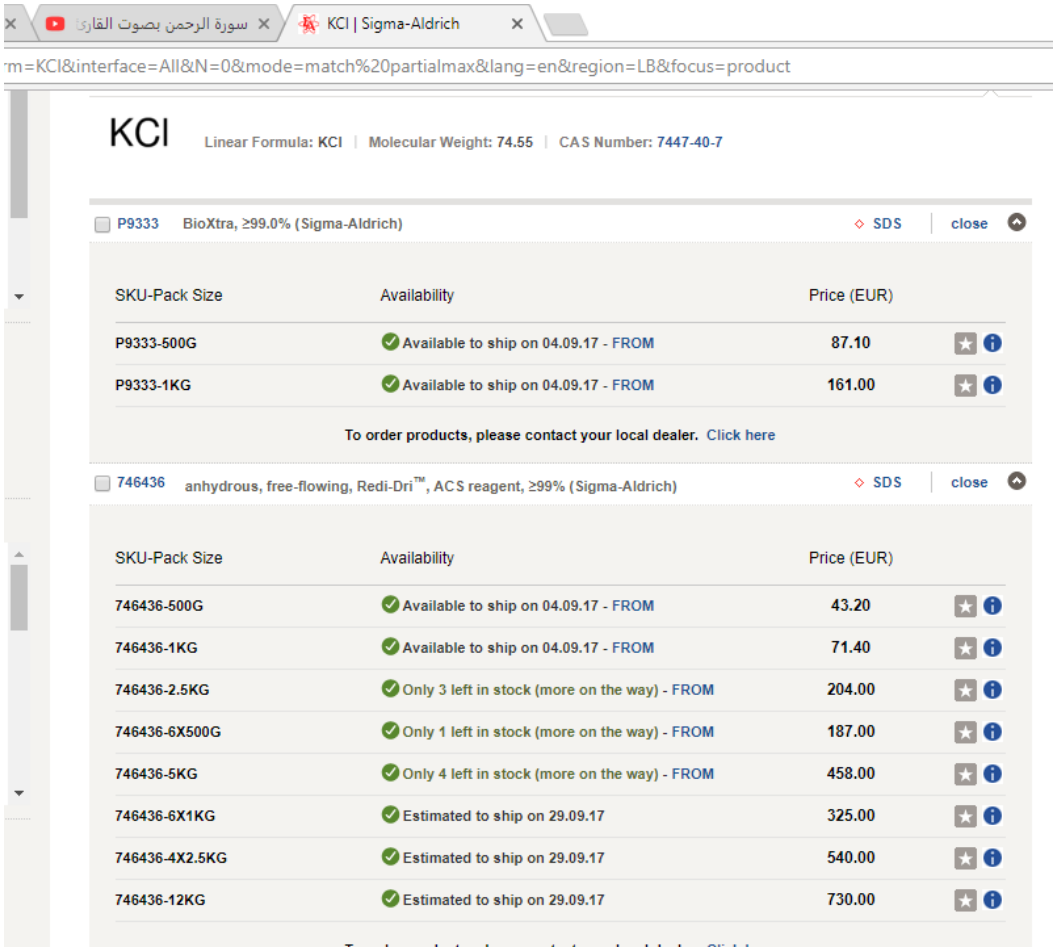


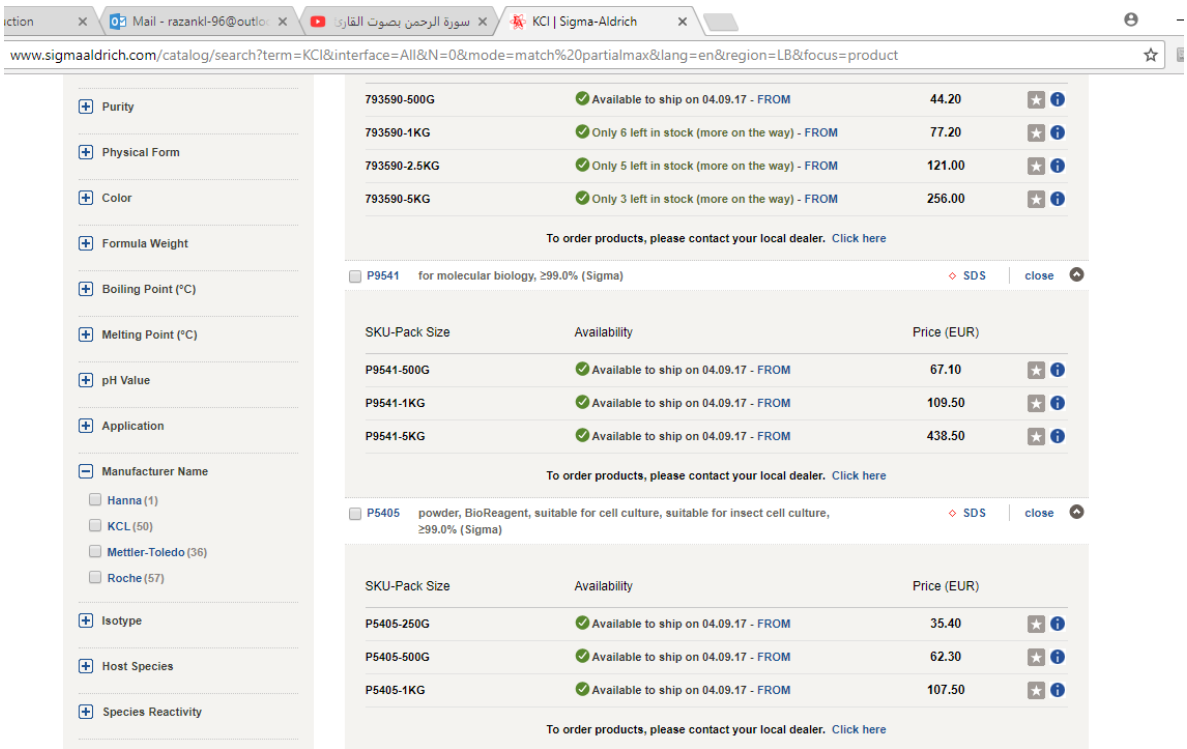
7.1.5 NaNO3:



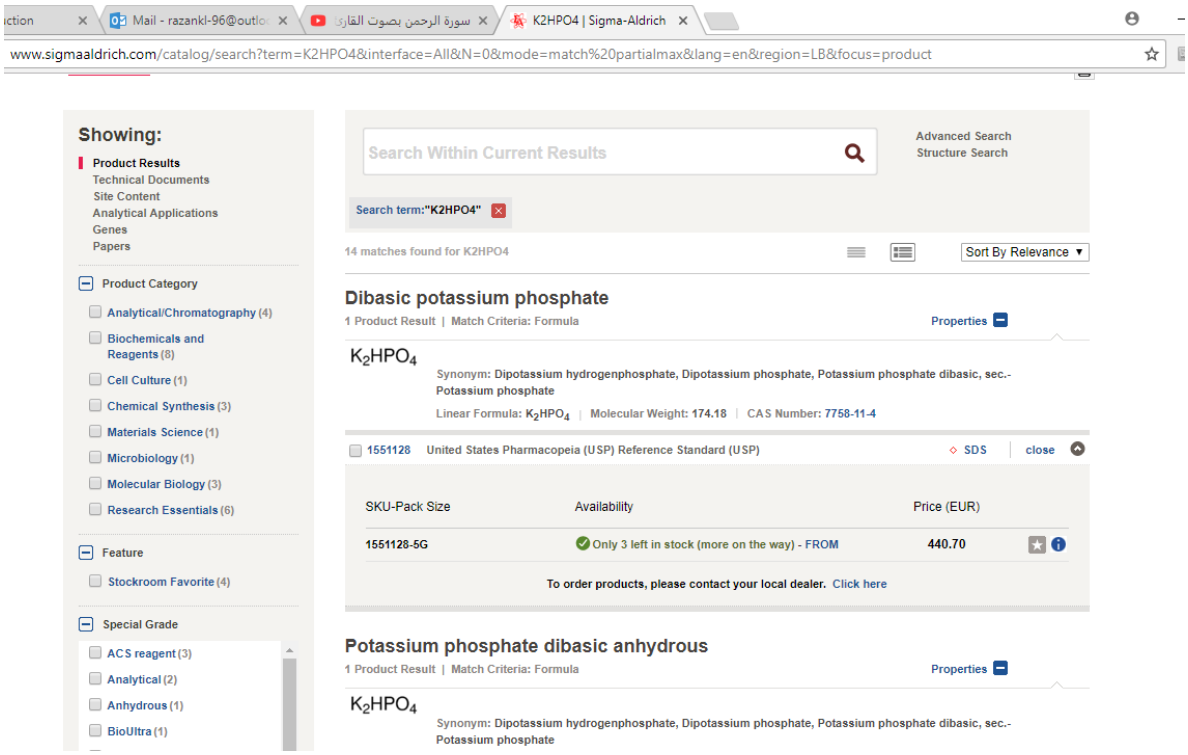

سورة الرحمن بصوت القارئ
NaNo3 | Sigma-Aldrich
rm=NaNo3&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product
15736 analytical standard (Sigma-Aldrich) SDS close
SKU-Pack Size Availability Price (EUR)
15736-1G Available to ship on 04.09.17 - FROM 31.40
To order products, please contact your local dealer. Click here
S8170 BioXtra, ≥99.0% (Sigma-Aldrich) SDS close
SKU-Pack Size Availability Price (EUR)
S8170-250G Available to ship on 04.09.17 - FROM 44.40
S8170-1KG Available to ship on 04.09.17 - FROM 141.00
To order products, please contact your local dealer. Click here
Show All 10 Results
Nitrogen and oxygen isotopes in nitrate
1 Product Result | Match Criteria: Formula
Properties
NaNO3
Synonym: Chile salpeter, Sodium nitrate
Linear Formula: NaNO3 | Molecular Weight: 84.99 | CAS Number: 7631-99-4
NISTRM8569 NIST® RM 8569 (Sigma-Aldrich) SDS close
SKU-Pack Size Availability Price (EUR)
سورة الرحمن بصوت القارئ
NaNo3 | Sigma-Aldrich
rm=NaNo3&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product
SKU-Pack Size Availability Price (EUR)
S8170-250G Available to ship on 04.09.17 - FROM 44.40
S8170-1KG Available to ship on 04.09.17 - FROM 141.00
To order products, please contact your local dealer. Click here
Show All 10 Results
Nitrogen and oxygen isotopes in nitrate
1 Product Result | Match Criteria: Formula
Properties
NaNO3
Synonym: Chile salpeter, Sodium nitrate
Linear Formula: NaNO3 | Molecular Weight: 84.99 | CAS Number: 7631-99-4
NISTRM8569 NIST® RM 8569 (Sigma-Aldrich) SDS close
SKU-Pack Size Availability Price (EUR)
NISTRM8569 Estimated to ship on 28.09.17 752.00
To order products, please contact your local dealer. Click here
SILu™ PrEST NANO3
1 Product Result | Match Criteria: Product Name, Property
QPREST39830 SILuPrESTs Powered by Atlas Antibodies, buffered aqueous solution (Sigma) SDS pricing

7.1.6 KCl


سورة الرحمن بصوت القارئ
KCl | Sigma-Aldrich
m=KCl&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product
KCl
Linear Formula: KCl | Molecular Weight: 74.55 | CAS Number: 7447-40-7
P9333 BioXtra, ≥99.0% (Sigma-Aldrich)
SDS
close
SKU-Pack Size
Availability
Price (EUR)
P9333-500G
Available to ship on 04.09.17 - FROM
87.10
P9333-1KG
Available to ship on 04.09.17 - FROM
161.00
To order products, please contact your local dealer. Click here
746436 anhydrous, free-flowing, Redi-Dri™, ACS reagent, ≥99% (Sigma-Aldrich)
SDS
close
SKU-Pack Size
Availability
Price (EUR)
746436-500G
Available to ship on 04.09.17 - FROM
43.20
746436-1KG
Available to ship on 04.09.17 - FROM
71.40
746436-2.5KG
Only 3 left in stock (more on the way) - FROM
204.00
746436-6X500G
Only 1 left in stock (more on the way) - FROM
187.00
746436-5KG
Only 4 left in stock (more on the way) - FROM
458.00
746436-6X1KG
Estimated to ship on 29.09.17
325.00
746436-4X2.5KG
Estimated to ship on 29.09.17
540.00
746436-12KG
Estimated to ship on 29.09.17
730.00


Mail - razankl-96@outlo
سورة الرحمن بصوت القارئ
KCl | Sigma-Aldrich
www.sigmaaldrich.com/catalog/search?term=KCl&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product
Purity
Physical Form
Color
Formula Weight
Boiling Point (°C)
Melting Point (°C)
pH Value
Application
Manufacturer Name
Hanna (1)
KCL (50)
Mettler-Toledo (36)
Roche (57)
Isotype
Host Species
Species Reactivity
793590-500G
Available to ship on 04.09.17 - FROM
44.20
793590-1KG
Only 6 left in stock (more on the way) - FROM
77.20
793590-2.5KG
Only 5 left in stock (more on the way) - FROM
121.00
793590-5KG
Only 3 left in stock (more on the way) - FROM
256.00
To order products, please contact your local dealer. Click here
P9541 for molecular biology, ≥99.0% (Sigma)
SDS
close
SKU-Pack Size
Availability
Price (EUR)
P9541-500G
Available to ship on 04.09.17 - FROM
67.10
P9541-1KG
Available to ship on 04.09.17 - FROM
109.50
P9541-5KG
Available to ship on 04.09.17 - FROM
438.50
To order products, please contact your local dealer. Click here
P5405 powder, BioReagent, suitable for cell culture, suitable for insect cell culture, ≥99.0% (Sigma)
SDS
close
SKU-Pack Size
Availability
Price (EUR)
P5405-250G
Available to ship on 04.09.17 - FROM
35.40
P5405-500G
Available to ship on 04.09.17 - FROM
62.30
P5405-1KG
Available to ship on 04.09.17 - FROM
107.50
To order products, please contact your local dealer. Click here

7.1.7 K2HPO4


K2HPO4 | Sigma-Aldrich
www.sigmaaldrich.com/catalog/search?term=K2HPO4&interface=All&N=0&mode=match%20partialmax&lang=en&region=LB&focus=product
Showing:
Product Results
Technical Documents
Site Content
Analytical Applications
Genes
Papers
Product Category
Analytical/Chromatography (4)
Biochemicals and Reagents (8)
Cell Culture (1)
Chemical Synthesis (3)
Materials Science (1)
Microbiology (1)
Molecular Biology (3)
Research Essentials (6)
Feature
Stockroom Favorite (4)
Special Grade
ACS reagent (3)
Analytical (2)
Anhydrous (1)
BioUltra (1)
Search Within Current Results
Advanced Search
Structure Search
Search term:"K2HPO4"
14 matches found for K2HPO4
Sort By Relevance
Dibasic potassium phosphate
1 Product Result | Match Criteria: Formula
Properties
K2HPO4
Synonym: Dipotassium hydrogenphosphate, Dipotassium phosphate, Potassium phosphate dibasic, sec.-Potassium phosphate
Linear Formula: K2HPO4 | Molecular Weight: 174.18 | CAS Number: 7758-11-4
1551128 United States Pharmacopeia (USP) Reference Standard (USP)
SDS
close
SKU-Pack Size
Availability
Price (EUR)
1551128-5G
Only 3 left in stock (more on the way) - FROM
440.70
To order products, please contact your local dealer. Click here
Potassium phosphate dibasic anhydrous
1 Product Result | Match Criteria: Formula
Properties
K2HPO4
Synonym: Dipotassium hydrogenphosphate, Dipotassium phosphate, Potassium phosphate dibasic, sec.-Potassium phosphate

8 Concept for production of Monoclonal Antibodies

8.1 Concept (from Dr Ahmad Trad)

8.1.1 Project Plan for Monoclonal antibody core facility

Monoclonal antibody

core facility

Dr.Ahmad Trad

04.06.2016

Summary of the project:

Antibodies are proteins generated by the adaptive immune system in response to the presence of an antigen. Antigen is considered any substance capable to elicit an immune response in the host. Antibodies are very stable molecules and bind with high specificity and affinity to target antigen. These features have made them one of the most attractive molecules in medicine biochemistry, and molecular biology. Antibodies are used extensively in therapy [1] and diagnosis of diseases, and in biomedical research. In fact, more than 20 monoclonal antibodies have been approved by the U.S. Food and Drug Administration for use in humans,

e. g, Heceptin, an anti-Her2 antigen monoclonal antibody, is used to treat metastatic breast cancer [2]. In addition, antibodies are also used in several diagnostic tests to detect small amounts of drugs, toxins or hormones, e.g. estimation of the hormone chorionic gonadotropin expressed during pregnancy and detection of HIV [3]. Furthermore, antibodies are vital tools for researchers to identify and to trace specific cells or molecules in an organism.

Antibodies are produced by B lymphocytes that specifically bind one antigen. Monoclonal antibody represents antibody generated from a single B cell. Consequently, monoclonal antibody recognises only one unique epitope on the antigen, which contains many epitopes. Monoclonal antibody producing B cells can be isolated from immunized animals and cultured in intro to produce monoclonal antibody. However, these B cells have a limited life span and produce a very low amount of antibody before they die. To overcome this problem, different techniques, including the hybridoma technology, have been developed to immortalized B cells. The hybridoma technology [4] involves the fusion of B cells of immunized animals with immortal myeloma cells, resulting in an immortalized cell line (Hybridoma) expressing antibodies of a defined specificity in the cell culture supernatant. This cell line allows production of monoclonal antibody in vitro in large quantity and when it is required. Hybridoma cells can be frozen and stored indefinitely at -80°C or in liquid nitrogen. Thus, hybridoma technology provides an immortal cell lines able to produce unlimited quantities of highly specific monoclonal antibodies.

As monoclonal antibodies are essential reagents for therapy, diagnostic of diseases and biomedical research, the increasing demand for developing new antibodies will continue. Indeed, monoclonal antibodies sector is fastest growing branch of biopharmaceutical industry.

As far as we know, neither in Lebanon nor in other arabic countries, exist biotechnology companies or academic institutes offering the generation of antibodies. Thus, there is need to establish monoclonal antibody core facility to provide the market with the inevitable reagents and meet the constantly rising demand for monoclonal antibodies. The goal of the core facility: (1) Transfer the hybridoma technology to Lebanon. (2) Training young researchers. (3) Production of monoclonal antibodies in mouse and rat to cover the need of the research centres and biotech companies for monoclonal antibodies

Aims of the project:

Scientific Goals	Making of antibodies for scientific research.
	Making courses and workshops for the young researches.
	Publish of high quality papers and reviews in international journals
	Hosting scientific conferences
Commercial Goals	Making ELISA kits
	Making Ag-specific hybridomas cell lines

Methods and technology:

Production of monoclonal antibodies involves *in vivo* (ascites) or *in vitro* procedures or combinations thereof. The first step for in both methods is to generate hybrid cells that are able to produce the antibodies. The steps in producing those cells are outlined below. The generation of mAb-producing cells requires the use of animals, usually mice. The procedure yields a cell line capable of producing one type of antibody protein for a long period. A tumor from this "immortal" cell line is called a hybridoma.

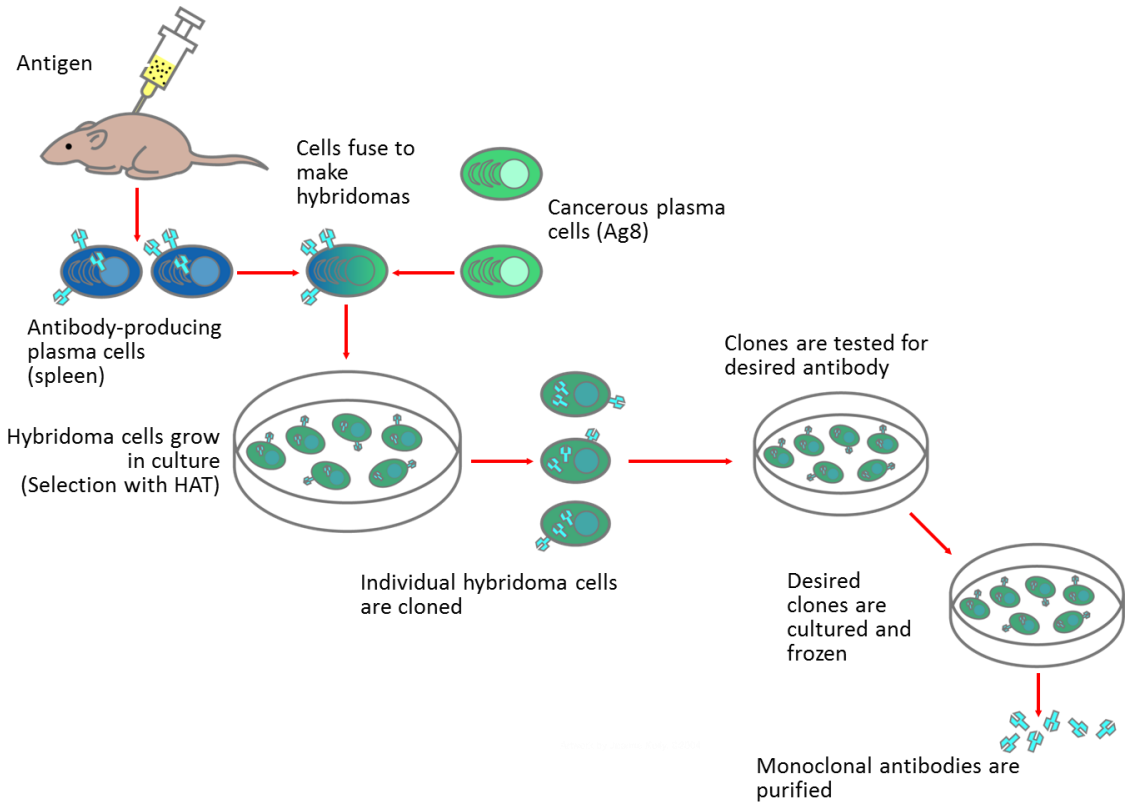


Fig 1. Hybridoma is a hybrid cell produced by injecting an antigen into a mouse, collecting an antibody-producing cell from the mouse's spleen, and fusing it with a long-lived cancerous immune cell called a

myeloma cell. Individual hybridoma cells are cloned and tested to find those produce the desired antibody. Cell supernatant of positive clone is collect and prepared for antibody purification.

It has been possible to genetically replace much of the mouse mAb-producing genes with human sequences, reducing the immunogenicity of mAb destined for clinical use in humans. Before the advent of the hybridoma method, investigators could produce only polyclonal serum antibodies; this required large numbers of immunized animals and did not immortalize the antibody-producing cells, so it required repeated animal use to obtain more antibodies. Development of the hybridoma technology has reduced the number of animals (mice, rabbits, and so on) required to produce a given antibody.

Step 1: Immunization of mice

Mice are immunized with an antigen that is prepared for injection either by emulsifying the antigen with Freund's adjuvant or other adjuvants or by homogenizing a gel slice that contains the antigen. Intact cells, whole membranes, and killed microorganisms are used as immunogens. In almost laboratories, mice are used to produce the desired antibodies. In general, mice are immunized every 2-3 weeks. However, the immunization protocols vary among investigators. When a sufficient antibody titer is reached in serum, immunized mice are euthanized and the spleen removed to use as a source of cells for fusion with myeloma cells.

Step 2: Screening of mice for antibody production

After few weeks of immunization, blood samples are collected from mice for measurement of serum antibodies. Several humane techniques have been developed for collection of small volumes of blood from mice. Serum antibody titer is determined with various techniques, such as enzyme-linked immunosorbent assay (ELISA) and flow cytometry. If the antibody titer is high enough, cell fusion can be performed. If the immune titer is too low, mice can be boosted until an adequate response is achieved, as determined by repeated blood sampling. When the antibody titer is high enough, mice are commonly boosted by injecting antigen without adjuvant intraperitoneally or intravenously (via the tail veins) 3 days before fusion. Then the mice are euthanized and their spleens removed for *in vitro* hybridoma cell production.

Step 3: Preparation of myeloma cells

Fusing antibody-producing spleen cells, which have a limited life span, with cells derived from an immortal tumor of lymphocytes (myeloma) results in a hybridoma that is capable of unlimited growth. A few weeks before cell fusion, myeloma cells must be treated against

mycoplasma as following: Tiamulin is added to a final concentration of 10 µg/ml for three days follows by minocycline at 5 µg/ml for 4 days. The treatment cycle is repeated three times. Cells must have high viability and rapid growth. The HAT medium allows only the fused cells to survive in culture.

Step 4: Fusion of myeloma cells with immune spleen cells

Single spleen cells from the immunized mouse are fused with the previously prepared myeloma cells. Fusion is accomplished by co-centrifuging freshly harvested spleen cells and myeloma cells in polyethylene glycol, a substance that causes cell membranes to fuse. As noted in step 3, only fused cells will grow in the special selection medium contained HAT, 10 % FCS and 10 % J774 supernatant. J774 Supernatant is believed to supply growth factors that promote growth of the hybridoma cells.

Step 5: Screening the hybridoma cell lines

At this step new, small clusters of hybridoma cells from the 96 well plates can be grown in tissue culture and the supernatant can be tested by ELISA followed by selection for positive clones. Cell supernatant of positive clone is collect and prepared for antibody purification.

Budjet of the project

Laboratory equipments	UV detector
	ELISA Reader
	Western blot equipment
	Inverted microscope
	Centrifuges for 1.5 ml and 50 ml
	Incubators
	Water bath
	Fume hood
	PCR machines
	Electrophoresis
	Refrigerators (-20 and -80 C)
	Autoclave
	Vacuum pump for filteration
	pH meter
Chemicals	RPMI 1640

	FCS
	Glutamine
	Penicillin sulfate salt
	Streptomycin sulfate salt
	Zellkulturschale 35 x 10 mm
	2-Mercaptoethanol
	Sodium bicarbonate NaHCO3
	Zellkulturplatte 24 Well
	HAT suplement
	HT suplement
	Polyethylenglycol PEG1500
	DMSO
	GERBU ADJUVANT MM
	Corning® cell strainer
	Mouse Restrainer
	Forceps/Scissors
	1 ml syringes
	Needle 27g
	ELISA Plates
	Goat Anti-Mouse Ig
	ABTS tablets
	Buffer for ABTS
	Casein
	Disodium carbonate (Na2CO3)
	Sodium chloride (NACL)
	Potassium chloride (KCl)
	Disodium hydrogen phosphate (Na2HPO4)
	Potassium dihydrogen phosphate (KH2PO4)
Animals	Balb/c mice
	Rats

Project team

	Degree/Name	POSITION
Project Team	DR. Ahmad Trad	Project manager
	Dip-Ing. Samir Mourad	Project Manager
	XX	Technical assistant

References

1. Sanchez-Carbayo, M., *Antibody Arrays: Technical Considerations and Clinical Applications in Cancer.* Clin Chem, 2006. **52**(9): p. 1651-1659.
2. Hillmen, P., et al., *Effect of the complement inhibitor eculizumab on thromboembolism in patients with paroxysmal nocturnal hemoglobinuria.* Blood, 2007. **110**(12): p. 4123-4128.
3. Wu, L.W., et al., *[OKT3 for recipients with steroid-resistant acute rejection after liver transplantation.].* Zhonghua Gan Zang Bing Za Zhi, 2007. **15**(11): p. 857-8.
4. Kohler, G. and C. Milstein, *Continuous cultures of fused cells secreting antibody of predefined specificity.* Nature, 1975. **256**(5517): p. 495-497.

8.1.2 Materials Needed

Materials needed for mAbs.xlsx

8.1.2.1 Equipment

Equipments
UV detector
ELISA Reader
Western blot equipment
Inverted microscope
Centrifuges for 1.5 ml and 50 ml
Incubators
Water bath
Fume hood
PCR machines
Electrophoresis
Refrigerators (-20 and -80 C)
Autoclave
Vacuum pump for filteration
pH meter

8.1.2.2 CHEMICALS

ID	Supplier	Cat. No.	Price in CZK	Price in USD	Amount	Note
Cell culture						
RPMI 1640	Sigma	R6504-50L	3712,5	158,9	50 L	Powder
FCS	Sigma	13063C-1000ML	2397,6	102,6	1 L	Australia origen
Glutamine	Sigma	G8540-25G	1449,9	62,1	25 G	
Penicillin sulfate salt	Sigma	S9137-25G	1247,4	53,4	25 G	
Streptomycin sulfate salt	Sigma	S6501-5G	540,0	23,1	5 G	
Zellkulturschale 35 x 10 mm	Sarstedt	83.3900.500	?	#WERT!	500 pieces	TC-Schale 35,Suspension
2-Mercaptoethanol	Sigma	M3148-25ML	464,4	19,9	25 ml	
Sodium bicarbonate NaHCO3	Sigma	S4019-500G	1609,2	68,9	500 G	
Zellkulturplatte 24 Well	Sarstedt	83.3922.005	?	#WERT!	500 pieces	TC-Platte 24 Well,Standard,F
HAT suplement	Sigma	H0262-10VL	3793,5	162,4	10 vials	
HT suplement	Sigma	H0137-10VL	3334,5	142,7		
Polyethylenglycol PEG1500	Sigma	P7306-5X5ML	3550,5	152,0	25 ml	
DMSO	Sigma	D2438-50ML	3726,0	159,5	50 ml	
GERBU ADJUVANT MM	Gerbu	30.011.000		130,0	1 ml	
Corning® cell strainer	Sigma	CLS431750-50EA	2.373,30	101,6	50 pieces	size 40 µm
Mouse Restrainer	Carlroth	PK28.1	71.95 Euro	80,7	1	
Forceps/Scissors	-	-	-	-	-	
1 ml syringes	-	-	-	-	-	
Needle 27g	-	-	-	-	-	

ELISA						
ELISA Plates	Sarstedt	82.1581.210	?	#WERT!	50 pieces	ELISA-Platte weiß High Bind. F
Goat Anti-Mouse Ig	Southernbiotech	1010-05	$115.00	115,0	1 ml	
ABTS tablets	Sigma	11 204 521 001	1674,0	71,6	20 tablets	pkg of 20 tablets
Buffer for ABTS	Sigma	11 204 530 001	1107,0	47,4	125 ml	
Casein	Sigma	C8654-500G	1323,0	56,6	500 g	
Disodium carbonate (Na2CO3)	Sigma	S7795-500G	2222,1	95,1	500 g	
Sodium chloride (NACL)	Sigma	S7653-250G	1142,1	48,9	250 g	
Potassium chloride (KCl)	Sigma	P9333-500G	2351,7	100,7	500 g	
Disodium hydrogen phosphate (Na2HPO4)	Sigma	S7907-100G	1512,0	64,7	100 g	
Potassium dihydrogen phosphate (KH2PO4)	Sigma	P5655-100G	726,3	31,1	100 g	

8.1.2.3 Cells, Animals

Animals	strain	Number	Price
Mouse	Balb/c	5	?

Cells	J774	Available
	Ag8	Available

8.1.2.4 Website

About Us	
Platform	The Monoclonal Antibody Core Facility provide biotech companies and research centers with high quality of monoclonal antibody using the hybridoma technology. Our scientists possess longstanding expertise and extensive know-how in generation of monoclonal antibody and establishing of ELISA. Our services include: • Advice on the design of Peptide/Protein for immunization • Production of mouse or rat monoclonal antibodies • Determination of Ig class and subclass • Production of monoclonal antibody in mg scale • Purification and labelling of antibodies • Long-term storage of hybridomas • ELISA development
Management	Dip-Ing. Samir Mourad › Read biography › Dr. Ahmad Trad › Read biography ›

Generation of Monoclonal Antibody

According to the hybridoma technology, the generation of monoclonal antibody included the following steps:

- Immunisation of 3-5 Mice/Rats
- Bleeding of the animals
- Titre determination in the antiserum by ELISA
- Preparing of the myeloma cell line, and isolation splenocytes from immunized animal after last booting
- Fusion of myeloma cells with splenocytes using PEG
- Selection of hybridomas in HAT medium
- Screening of hybridomas for antigen specificity
- Cloning of positive clone by limiting dilution
- Determination of Ig subclass
- Delivery of cell culture supernatants
- Antibody purification from supernatant by Protein A/G (2 clones, 3mg each)
- Cryo conservation of the

positive clones (each 3 cryo vials)

- Delivery of final report to customer

It is always helpful to discuss with us in detail the design of your project to adapt all above mentioned steps to fulfill your expectation of the generated monoclonal antibody.
Clones and correspond obtained results are the property of the customer.
Please contact us for a detailed quotation.

Establishing ELISA Kit for detection of antigen ›

Establishing ELISA Kit for detection of antigen

In order to quantify your antigen, we offer searching for a pair of monoclonal antibodies (one capturing and one for detection) suitable to be used in a sandwich ELISA. Furthermore, we offer establishing of complete ELISA kit including labeling of detection antibody. The complete ELISA kit contains the following reagents:

- Purified primary antibody for capturing of antigen
- Biotin labeled secondary antibody for detection
- Sufficient amount of coating, washing and detection buffers

Purification of antibody from established hybridomas ›

Purification of antibody from established hybridomas

For hybridomas provided by customer, we offer the following services:

- Cultivation of hybridomas

- Cloning by limiting dilution
- Determination of Ig subclass
- Long time storage of clones
- Production sufficient amount of cell supernatant as requested by customers
- Purification of monoclonal antibody in various scales from (3 mg per batch)
- Affinity determination

Price ›

8.1.2.5 Products and price

Description	**Cat. No.**	**Price, USD**
Immunisation of 3-5 Mice/Rats	AT-10	10000
Titre determination in the antiserum by ELISA		
Fusion of myeloma cells with splenocytes using PEG		
Screening of hybridomas for antigen specificity		
Cloning of positive clone by limiting dilution		
Determination of Ig subclass		
Antibody purification from supernatant by Protein A/G (2 clones, 3 mg each)		
Cryo conservation of the positive clones (each 3 cryo vials)		
Establishing of ELISA Kit for detection of antigen	AT-11	20000
Cloning by limiting dilution	AT-12	1000
Determination of Ig subclass	AT-13	1000
Purification of monoclonal antibody in various scales from (3 mg per batch)	AT-14	2000
Affinity determination	AT-15	1000
Biotinilation of antibody	AT-16	1000

8.2 Email Correspondence with Dr Ahmad Trad concerning business concept

8.2.1 Introduction and meeting in Nuremberg in Ramadan 2017 (first Ramadan week)

Re: Transfer of Hybridoma Technology

8. Juni 2017 | 01:00 | 519 KB

Von:
Ahmad Trad <ahmadtrad24@yahoo.de>

An:
samir.mourad@aecenar.com

Lieber Bruder Samir,
Aslam Alikum.
Zuerst möchte ich mich bei Dir für den warmen Empfang bedanken, das Treffen hat mich viele Freude gebracht. Es freute mich sehr, Dir als neuer Bruder kennen zu lernen.
Transferieren die Hybridoma Technologie zu unserem Land (Uma), find ich ganz wichtig. Er ist unser gemeinsamer Ziel und das sollen wir im Auge behalten.

Zu deiner Business Strategie, wenn ich richtige verstanden habe, sieht es so aus:

Das Labor steht bereit zu Verfügung

1) Räume sollten vermietet werden (ca. 500 $/Monat). Hier wurde gewünscht, dass ich die Kosten übernehme. Das Geld wird als Investition gerechnet, z.B. 6000 $ pro Jahr sind ca. 6% von dem gesamter Investition, die ca. 100.000 $ beträgt. d.h. mein Teil von Gewinn der Firma ist 6%.
2) Mein Know-how wird als Arbeitsstunden gerechnet. Es wird pro Stunde bezahlt.
3) Ein Technischer Assistent für das Labor wird irgendwie besorgt (Masterstudent oder Ähnlicher)

Momentan leider von der Finanzierung her, bin ich nicht in der Lage das Geld zu investieren. Ich weiß das ist nicht viel (6000 $ pro Jahr) aber jede von uns hat eine bestimmte Kapazität. Ich weiß das etwas für die Uma, aber wirklich meine Situation erlaubt das nicht.

Mein Vorschlag wäre, ihr übernimmt das gesamte kosten und ich übernehme die unten aufgelisteten Aufgaben:

1) Training technischen Assistenten
2) Täglich Kontakt mit TA über das verläuft der Arbeit
3) Durchführung Kontakt mit Kunden und Projekt Planung am Samstags und Sonntags

So sind meine Aufgaben, ich werde mein Best tun um die Firma auf eigene Füße stehen zu können. Ich bin überzeugt, davon eine hervorragende Arbeit entsteht.
Was ich dafür verlange:
1) So lange die Firma kein Gewinnt macht, brauche ich kein Geld.
2) Wenn die Firma Gewinnt macht, dann würde ich gerne als co-founder 25% haben.

Ich weiß das ist eine Arbeit für die Uma, und ich tue mein Best so lange ich kann.

Ich hoffe mein Vorschlag gefällt Dir. Auf jeden Fall bin ich offen und bereit andere Möglichkeiten zu diskutieren. Ich habe nicht vergessen, dass dieser Arbeit inscha Allah für die Uma gewidmet ist, aber jede von uns hat seine grenze.
Anbei findest du einige Information über die Produktion von monoklonalen Antikörper.
Diese Informationen sollten bitte als vertraulich behandelt werden.
Ich danke Dir noch mal, und hoffe bald von Dir zu hören.

Mit freundlichen Grüßen,

Ahmad

Dr. rer. nat. Ahmad Trad
Diplom Biochemistry
Waitzstraße 68
24105 Kiel
Germany

Von: Samir Mourad <samir.mourad@aecenar.com>
An: Ahmad Trad <ahmadtrad24@yahoo.de>
Gesendet: 18:02 Sonntag, 14.Mai 2017
Betreff: Re: Transfer of Hybridoma Technology

Sehr geehrter Dr Ahmad
as Salamu alaikum,

es freut mich, dass Sie an uns wenden wegen Ihrem Projekt.
Wir können gerne über weitere Einzelheiten sprechen.
Wie ich es verstanden habe, brauchen Sie nur ein Labor. Suchen Sie auch eine Anstellung oder wollen Sie sich in Nordlibanon selbstständig machen?

Mit freundlichen Grüßen
Samir Mourad

Eng. Samir Mourad, Director
Phone (Mobile Lebanon) +961 76 341 526
(Mobile Germany) +49 (0)178 72 855 78
Email: samir.mourad@aecenar.com
Association for Technological and Economical Cooperation in the Euro-Asian and North-African Region (AECENAR)
Ras Nhache/Batroun
Lebanon

www.aecenar.com

Am 12. Mai 2017 um 16:39:59 +03:00, hat Ahmad Trad <ahmadtrad24@yahoo.de> geschrieben:

Dear Mr. Samir Mourad,

First of all Iwant to introduce myself. I am original from north Lebanon.
I earned my Ph.D. in Immunology in 2009 at Kiel University. From 2009 to 2014 I worked as a post-doctoral researcher, managed and supervised (Head) the antibody core facility at the biochemical Institute of Kiel University. Since 2014 I am the head of cell culture by Abcheck (Czech Republic).
I am planning to establish a core facility for monoclonal antibodies production. Despite my knowledge, I have the cells that allow me to produce such molecules. This project has many important approaches:

1- Important Technology Transfer to Lebanon (Hybridoma Technology)
2- Produce of antibodies for medical diagnostic (Medical approaches)
3- Produce of antibodies for scientific research

Required: Now I need biosafety level 1 laboratory in AECENAR Center Lebanon.

Please let me know if you need detailed information.

Best regards,

Ahmed Trad

Dr. rer. nat. Ahmad Trad
Diplom Biochemistry
Waitzstraße 68
24105 Kiel
Germany

Anhänge: Alle Anhänge herunterladen

Project Plan,05.06.2....doc

Materials needed fo....xlsx

8.2.2 Final result of discussion

Re: Transfer of Hybridoma Technology
12. Juni 2017 | 21:36 | 111 KB

Von:
Ahmad Trad <ahmadtrad24@yahoo.de>

An:
samir.mourad@aecenar.com

Lieber Bruder Samir,
ich wusste nicht dass du bzw. wir ein Produktionsstätte aufbauen willst, du hast mir nur kurz darüber gesprochen, dass in Libanon die meisten Firme mit Verpackungen von Medikament beschäftigt sind aber nicht mit Herstellung. Ich finde die Idee sehr gut. Denkst du ist es mögliche ein Dienstleistungsunternehmen (was ich geplant habe) mit Medikamentenproduktionsfirma in eurem facility Kombinieren werden könnte? Im Prinzip willst du ein Dienstleistungsunternehmen und Medikamentenproduktionsfirma aufbauen oder nur Medikamentenproduktionsfirma? Aber wie du in dein E-Mail erwähnt hast, dass ein schwieriger Weg ist. Hast du ein Produkt (sein Patent abgelaufen ist), Pläne für die Produktion? Vielleicht könnte ich dabei helfen, wenn du willst.
Wie auch immer, ich hoffe, dass wir irgendwie jetzt oder später zusammen arbeiten können. Auf jeden Fall, stelle ich Zellen, die man für Hybridoma Technologie braucht, zur deiner Verfügung, und biete ich (1-2 Tage) einen praktischen/theoretischen Kurs in der Herstellung von monoklonalen Antikörper an.

Möge Allah deine guten Taten vielfachen,
wa salamu alikum,
dein Bruder Ahmad

Dr. rer. nat. Ahmad Trad
Diplom Biochemistry
Waitzstraße 68
24105 Kiel
Germany

Von: Samir Mourad <samir.mourad@aecenar.com>
An: Ahmad Trad <ahmadtrad24@yahoo.de>
Gesendet: 22:12 Samstag, 10.Juni 2017
Betreff: Re: Transfer of Hybridoma Technology

As Salamu alaikum, lieber Bruder Ahmad

Wie ich verstanden habe, möchtest du ein Dienstleistungsunternehmen machen, ich möchte aber eine Produktionsstätte.
Bzgl. Dienstleistung haben wir jahrelange Erfahrung u.a. im Softwarebereich und da kann ich dir sagen, dass das zwar fast cash bringt, aber keine dauerhafte Firma im Sinne einer Medikamentenproduktionsfirma, da man sich immer an den kurzfristigen Wünschen der Kunden orientieren muss.

Ich denke, dass es eben besser ist, zunächst eine facility aufzubauen, und dann mit vorgefertigten Produkten auf den Markt zu kommen.
Es ist eben ein schwieriger Weg, aber der baut wirklich Technologie und Infrastruktur auf.
Khair inscha Allah.
Allah möge dich bewahren.

Wassalam,
dein Bruder Samir

Eng. Samir Mourad, Director
Phone (Mobile Lebanon) +961 76 341 526
(Mobile Germany) +49 (0)178 72 855 78
Email: samir.mourad@aecenar.com
Association for Technological and Economical Cooperation in the Euro-Asian and North-African Region (AECENAR)
Qubaisi Center
Ras Nhache/Batroun
Lebanon

www.aecenar.com

Am 10. Juni 2017 um 21:11:35 +02:00, hat Ahmad Trad <ahmadtrad24@yahoo.de> geschrieben:

> Lieber Bruder Samir,
>
> danke für die Erklärung, es ist mir jetzt sehr klar.
>
> Wahrscheinlich wir reden über zwei unterschiedlichen Sachen was die Fima umgeht.
> Die Firma, die ich mir vorgestellt habe, ist ein Service Firma. D.h wir bekommen Aufträge von „Kunden" (Universitäten, Forschungszentrum, Firmen) um Antikörper gegen ihrem Antigen herzustellen. Mit unserer Hybridoma Technologie, werden wir monoklonalen Antikörper gegen das Antigen produzieren. Das ist die einfachste Version als Start Punkt für unser Projekt.
>
> Wenn ich dir richtig verstanden habe, die Firma soll ein bestimmt typ von monoklonalen Antikörper (z.b. biosimilare Antikörper, Infliximab) produzieren. Dafür brauchst du aber nicht die Hybrydoma Technologie.
>
> Eine Zweite Punkt, nach meiner Vorstellung, wir brauchen keinen Prototyp, oder es gibt keinen Prototyp wenn wir die Hybrydoma Technologie benutzen. D.h für jeden Auftrag (Antigen), starten wir neuen versuch. Der erste Auftrag wäre so zu sagen unser prototyp.

Ich denke was wir machen könnten, versuchen wir Auftrage zu bekommen, bevor wir das Labor öffnen oder in parallel. D.h wir sollten uns an Universitäten, Firmen und Krangenhause vorstellen und sagen wir haben (oder in kurze Zeit) diese Technologie und ob sie daran Interesse haben.Man könnte Anzeige an eurem Website machen, dass ein "Antibodies core facility" under construction ist.

Ich denke es ist wichtige zu vereinigen, was prinzipiell die Firma produzieren soll. Das Hybrydoma Technologie zu haben ohne Aufträge von „Kunden" macht es wenig Sinn. Weil wir momentan kein eigenes Target (Antigen) haben, das wir daran arbeiten können.

Bitte überleg es dir, was ich oben erwähnt habe. Schreib mir wie du die Firma vorstellst und was sie produzieren soll. Es wäre natürlich gut wenn du mir auch eine Zusammenfassung über die Finanzierung der Firma schickt, wer wieviel bezahlt, was es fehlt, vielleicht könnte ich Investoren finden, die diese Arbeit unterstützen.
Von meiner Seite, ich werde Kollege, die in arabischen Ländern an der Universität unterrichten, kontaktieren, ob sie potentiale „Kunden" für uns kennen.

Nach deinem Bussnis Modul, kann ich leider nur Arbeitskraft investieren.

Wa assalam alikum,

dein Bruder Ahmad

Dr. rer. nat. Ahmad Trad
Diplom Biochemistry
Waitzstraße 68
24105 Kiel
Germany

Von: Samir Mourad <samir.mourad@aecenar.com>
An: Ahmad Trad <ahmadtrad24@yahoo.de>
CC: "rs130893@gmail.com" <rs130893@gmail.com>; Samir Mourad <smourad69@googlemail.com>
Gesendet: 15:11 Samstag, 10.Juni 2017
Betreff: Re: Transfer of Hybridoma Technology

Wa alaikum as Salam, achi Ahmad

zu deinen Fragen:
1. jeder kann mit Geld und Arbeitskraft investieren. Dafür bekommt man Anteile (Aktien). Wenn du dich entschieden hast, mitzumachen, legen wir zusammen die Gesamtinvestitionssumme für die Laufzeit der Firma fest.
Die Investitionssumme sollte reichen, dass wir einen Prototyp machen und auf den Markt kommen, so dass die return of invest phase startet
2. Manche Leute bezahlen wir gleich, wie z.B. wenn wir einen Elektriker brauchen oder ein Vermieter, wenn wir seine Wohnung für die Firma brauchen

3. Natürlich soll die Firma gewinnbringend werden. Der erste Schritt ist aber immer, einen Prototyp herzustellen, in diesem Fall ein Prototyp für die Herstellung von hybrodoma antibodies. Um kosten zu sparen, machen wir dies am besten mit Studenten. Wenn jemand von uns Betreuungsleistung erbringt, bekommt er dafür Anteile entsprechend dem Stundenaufwand.
Vorher wurde natürlich im Businessplan festgelegt, dass dieses Arbeitspaket (Material+Personal) soundsoviel kostet (z.B. 20.000 EUR). Du betreust jetzt z.B. bei diesem Arbeitspaket einen Monat lang, vollzeit dann bekommst du 160h x dein Stundensatz (z.B. 80 EUR) = 12.800 EUR
Wenn wir z.B. als Gesamtinvestitionssumme festgelegt haben, dass die Firma 1.280.000 EUR wert ist (Kosten = Miete + Materialkosten + Personal für z.B. 3 Jahre), dann hast du 1% Anteile. Sobald dann die Firma Gewinn macht, bekommst du 1% vom Gewinn.
Schreib mir einfach über whatsapp, wenn noch Klärungsbedarf ist.
Baraka Allahu fik.

Wassalam,
Samir

Eng. Samir Mourad, Director
Phone (Mobile Lebanon) +961 76 341 526
(Mobile Germany) +49 (0)178 72 855 78
Email: samir.mourad@aecenar.com
Association for Technological and Economical Cooperation in the Euro-Asian and North-African Region (AECENAR)
Qubaisi Center
Ras Nhache/Batroun
Lebanon

www.aecenar.com

Am 10. Juni 2017 um 14:09:55 +02:00, hat Ahmad Trad <ahmadtrad24@yahoo.de> geschrieben:

> Lieber Bruder Samir,
> Assalam Alikum, vielen Dank für deine schnellere Antwort.
>
> Zu deinem Bussnis Modul, es ist mir eigentlich nicht klare, wie du die Sachen vorstellst. Kannst du mir bitte mehr erklären. Zum Beispiel wenn ich Geld und Arbeitskraft investiere, was bekomme ich konkret?
>
> a. Gute Taten, weil dieser Arbeit für die Uma gewidmet ist
> b. Ein Teil (25 %) von der Firm, wie ich vorgeschlagen habe
> c. Bezahlt pro Stunde, wann und wer bezahlt?
> d. Erfahrung wie ist der Markt, damit ich später meine eigene Firma aufbauen kann.

8.2.3 CV of Dr Ahmad Trad

Curriculum Vitae

Personal Information

Name: Ahmad Trad

Address:

Nepomucká 312/156

32600 Plzeň

Czech Republic

Telephone: +420 778019508

Email: ahmadtrad24@yahoo.de

Education:

2010-2014: Post-doctoral Fellow

Christian Albrecht University of Kiel, Department of Biochemistry

Advisor: Professor Dr. Joachim Grötzinger

2006-2010: Doctor of Philosophy in Biochemistry/Immunology

Christian Albrecht University of Kiel, Department of Biochemistry

Thesis: *Significance of the third hypervariable region of the antibody H chain for antigen-specificity and expression of idiotypes during the thymus-dependent immune response*

Advisor: Professor Dr. Hilmar Lemke (Immunology)

2003-2005: Master of Biochemistry/Molecular Biology

Christian Albrecht University of Kiel

Thesis: *Affinity determination of syngenic anti-idiotypic antibody using surface plasmon resonance and ELISA*

Advisor: Professor Dr. Hilmar Lemke (Immunology)

1998-2002: Bachelor of Science in Biochemistry

Lebanese University

Research Experience

2014 - Present: Head of cell culture at AbCheck s.r.o

- Managed projects of external customers
- Engineered transgenic animals
- Developed and optimized innovative technology
- Generation of stable cell lines (from e.g. CHO, HEK cell lines)

2010 - 2014 Post-doctoral Fellow

Christian Albrecht University of Kiel, Department of Biochemistry

- Conducted research in structural biology of ADAM metalloprotease-17 (ADAM17)
- Managed monoclonal antibody core facility (Head)
- Validated ADAM17 as potential target for immunotherapy
- Generated monoclonal antibodies against different antigens
- Developed sandwich ELISA to quantify ADAM17, UCH-L1, NOD2 and meprin α

2006 - 2010: Graduate Researcher

Christian Albrecht University of Kiel, Department of Biochemistry

- Investigated the immune response of transgenic mice harbored only one single DH gene in their genome instead of 13 DH genes as in BALB/c wild-type mice using 2-phenyloxazolone (phox) as model antigen
- Analyzed the immune response (Titer, B and T cells)
- Generated monoclonal antibodies against phox
- Executed bioinformatic analysis of VH/VL genes

Technical skills

- Generation of monoclonal antibodies by Hybridoma technology
- Engineering of transgenic animals
- Handling, immunization and bleeding of mice/rats (FELASA B)
- Generation of stable cell lines (from e.g. CHO, HEK cell lines)
- Protein/antibody engineering
- Molecular modeling of antibody V-regions
- DNA and protein sequence analysis software
    - Vector NTI, BioEdid, Chroma, Blast, Clustal omega

Computer Skills

Excel, Access, PowerPoint, Word, Photoshop, Windows, EndNote, Graphpad

Language Skills

German: Fluent (speaking, reading, writing)

English: Fluent (speaking, reading, writing)

Concept for production of Monoclonal Antibodies

Arabic:	Native
French:	Intermediate
Italian:	Basic

Publications

1) Hinrich P Hansen, **Ahmad Trad**, Paola Zigrino, Marcia Moss, Gisela Schön, Patricia C Grenzi, Bruno Aquino, Horst Dürkop, Katrin S Reiners, Michael Hallek, Achim Groetzinger, Adriana F Paes Leme, and Elke Pogge von Strandmann "Shedding of signaling proteins limits the functionality of cancer cell-derived extracellular vesicles to the tumor microenvironment in Hodgkin's lymphoma" Oncotarget. 2016 Apr 20. doi: 10.18632/oncotarget.8864.

2) Jin Qu, Hongtao Yu, Fenge Li, Chunlei Zhang, **Ahmad Trad**, Cory Brooks, Bin Zhang, Ting Gong, Zhi Guo, Yunsen Li, Yanyan Lou, Patrick Hwu, Wei Huang, Dapeng Zhou "Molecular basis of antibody binding to mucin glycopeptides in lung cancer" International Journal of Oncology 48, no. 2 (2016): 587-594. doi: 10.3892/ijo.2015.3302.

3) **Ahmad Trad**, Radu Iulian Tanasa, Tobias Rogosch, Michael Zemlin, Harry W. Schroeder Jr. and Hilmar Lemke "Clonal progression during the thymus-dependent B cell response to the hapten oxazolone depends on the immunoglobulin DH gene repertoire" Front Immunol. 2014 Aug 11;5:385. doi: 10.3389/fimmu.2014.00385.

4) Jan Sommer, Christoph Garbers, Janina Wolf, **Ahmad Trad**, Jens M. Moll, Joachim Grötzinger, Georg H. Waetzig, Doreen M. Floss and Jürgen Scheller "Alternative intronic polyadenylation generates the interleukin-6 trans-signaling inhibitor sgp130-E10" J Biol Chem. 2014 Aug 8;289(32):22140-50. doi: 10.1074/jbc.M114.560938.

5) Justyna Sosna, Susann Voigt, Sabine Mathieu, Dieter Kabelitz1, **Ahmad Trad**, Andreas Tholey, Ottmar Janssen, Catherine Meyer-Schwesinger, Stefan Schütze, and Dieter Adam "The proteases HtrA2/Omi and UCH-L1 regulate TNF induced necroptosis" Cell Communication and Signaling, 2013, 11:76 doi:10.1186/1478-811X-11-76.

6) Schweigert O, Dewitz C, Möller-Hackbarth K, **Ahmad Trad**, Garbers C, Rose-John S, Scheller J. "Soluble T cell immunoglobulin and mucin domain (TIM)-1 and -4 generated by a disintegrin and metalloproteinase domain-containing proteins (ADAM)-10 and -17 bind to phosphatidylserine" Biochim Biophys Acta. 2014 Feb;1843(2):275-87. doi: 10.1016/j.bbamcr.2013.11.014.

7) Katja Möller-Hackbarth1, Christin Dewitz, Olga Braun, **Ahmad Trad**, Marcia L. Moss, Christoph Garbers, Stefan Rose-John, and Jürgen Scheller "ADAM10 and ADAM17 are major sheddases of t cell immunoglobulin and mucin domain 3 (TIM-3)" The Journal of J Biol Chem. 2013 Nov 29;288(48):34529-44. doi: 10.1074/jbc.M113.488478.

8) Mirja Oldefest, Jana Kieper, Chien-Wen Hung, Denis Neelsen, **Ahmad Trad**, Andreas Tholey, Joachim Grötzinger, Inken Lorenzen "Upd3 - an ancestor of the four-helix bundle cytokines?" Biochem Biophys Res Commun. 2013 Jun 21;436(1):66-72. doi: 10.1016/j.bbrc.2013.04.107.

9) **Ahmad Trad**, Michel Riese, Mohammad Shomali, Nina Hedemana, Joachim Grötzinger Inken Lorenzen "The disintegrin domain of ADAM17 antagonises fibroblast-carcinoma cell interactions" Int J Oncol. 2013 May;42(5):1793-800. doi: 10.3892/ijo.2013.1864.

10) **Ahmad Trad**, Hinrich P Hansen, Mohammad Shomali, Matthias Peipp, Katja Klausz, Nina Hedemann, Kosuke Yamamoto, Christine Desel, Inken Lorenzen, Hilmar Lemke, Stefan Rose-John, Joachim Grötzinger "ADAM17-overexpressing breast cancer cells selectively targeted by antibody-toxin conjugates" Cancer Immunol Immunother. 2013 Mar;62(3):411-21. doi: 10.1007/s00262-012-1346.

11) Kosuke Yamamoto, **Ahmad Trad**, Anja Baumgart, Linda Hüske, Inken Lorenzen, Athena Chalaris, Joachim Grötzinger, Tobias Dechow, Jürgen Scheller and Stefan Rose-John "A novel bispecific single-chain antibody for ADAM17 and CD3 induces T-cell mediated lysis of prostate cancer cells" Biochem J. 2012 Jul 1;445(1):135-44. doi: 10.1042/BJ20120433.

12) Inken Lorenzen, Juliane Lokau, Stefan Düsterhöft1, Ahmad Trad, Christoph Garbers, Jürgen Scheller, Stefan Rose-John, Joachim Grötzinger "The membrane-proximal domain of A Disintegrin And Metalloprotease 17 (ADAM17) is responsible for recognition of the Interleukin-6 receptor and Interleukin-1 receptor II" FEBS Lett. 2012 Apr 24;586(8):1093-100. doi: 10.1016/j.febslet.2012.03.012.

13) Hans Lange, Oliver Hecht, Michael Zemlin, **Ahmad Trad**, Radu Iulian Tanasa, Harry W. Schroeder Jr and Hilmar Lemke "Immunoglobulin class switching appears to be regulated by B-cell antigen receptor-specific T-cell action" Eur J Immunol. 2012 Apr;42(4):1016-29. doi: 10.1002/eji.201141857.

14) Inken Lorenzen, **Ahmad Trad**, Joachim Grötzinger

"Multimerisation of A disintegrin and metalloprotease protein-17 (ADAM17) is mediated by its EGF-like domain" Biochem Biophys Res Commun. 2011 Nov 18;415(2):330-6. doi: 10.1016/j.bbrc.2011.10.056.

15) Abdelsadik A, **Ahmad Trad**

"Toll-like receptors on the fork roads between innate and adaptive immunity"

Hum Immunol. 2011 Dec;72(12):1188-93. doi: 10.1016/j.humimm.2011.08.015.

16) **Ahmad Trad**, Nina Heddemen, Verena Pawlak, Mohammad Shomali, Joachim Grötzinger, Inken Lorenzen " Development of sandwich ELISA for detection and quantification of human and murine a disintegrin and metalloproteinase17" J Immunol Methods. 2011 Aug 31;371(1-2):91-6. doi: 10.1016/j.jim.2011.06.015.

17) Lothar Marischen, Daniela Wesch, Hans-Heinrich Oberg, Philip Rosenstiel, **Ahmad Trad**, Mohammad Shomali, Joachim Grötzinger, Ottmar Janssen, Vladimir Tchikov, Stefan Schütze, Dieter Kabelitz

"Functional expression of NOD2 in human peripheral blood Gamma/Delta T-cells" Scand J Immunol. 2011 Aug;74(2):126-34. doi: 10.1111/j.1365-3083.2011.02560.

18) Radu I. Tanasa, **Ahmad Trad,** Hans Lange, Joachim Grötzinger, and Hilmar Lemke. ''Allergen IgE-isotype-specific suppression by maternally derived monoclonal anti-IgG-idiotype'' Allergy. 2010 Jan;65(1):16-23. doi: 10.1111/j.1398-9995.2009.02104.

19) Hilmar Lemke, Radu Iulian Tanasa, **Ahmad Trad** and Hans Lange "Benefits and burden of the maternally-mediated immunological imprinting'' Autoimmun Rev. 2009 Mar;8(5):394-9. doi: 10.1016/j.autrev.2008.12.005.

20) Hans Lange, Michael Zemlin, Radu Iulian Tanasa, **Ahmad Trad**, Thomas Weiss, Hauke Menning, Hilmar Lemke. ''Thymus-independent type 2 antigen induces a long-term IgG-related network memory'' Mol Immunol. 2008 May;45(10):2847-60. doi: 10.1016/j.molimm.2008.01.020.

Books

1) Hilmar Lemke, Radu Iulian Tanasa, **Ahmad Trad** and Hans Lange (2008). Immunolophysiology: Natural autoimmunity in physiology and pathology, P 77-86.

2) Hilmar Lemke, Radu Iulian Tanasa, **Ahmad Trad** and Hans Lange (2012). Maternal Fetal Transmission of Human Viruses and their Influence on Tumorigenesis. Springer, ISBN 978-94-007-4215-4

Selected Poster

Mass Humanization of Rabbit Antibodies

Jacob Glanville (Distributed Bio), Peter Milovník (AbCheck), Remko Griep (AbCheck), **Ahmad Trad** (AbCheck) and Vera Molkenthin (AbCheck), pegs_europe_2015.

Patents

1) Isolation of anti-Tyrosine-protein kinase transmembrane receptor ROR1 monoclonal antibodies: In Progress
2) Mass humanization of rabbit antibodies: In progress

9 Suppliers

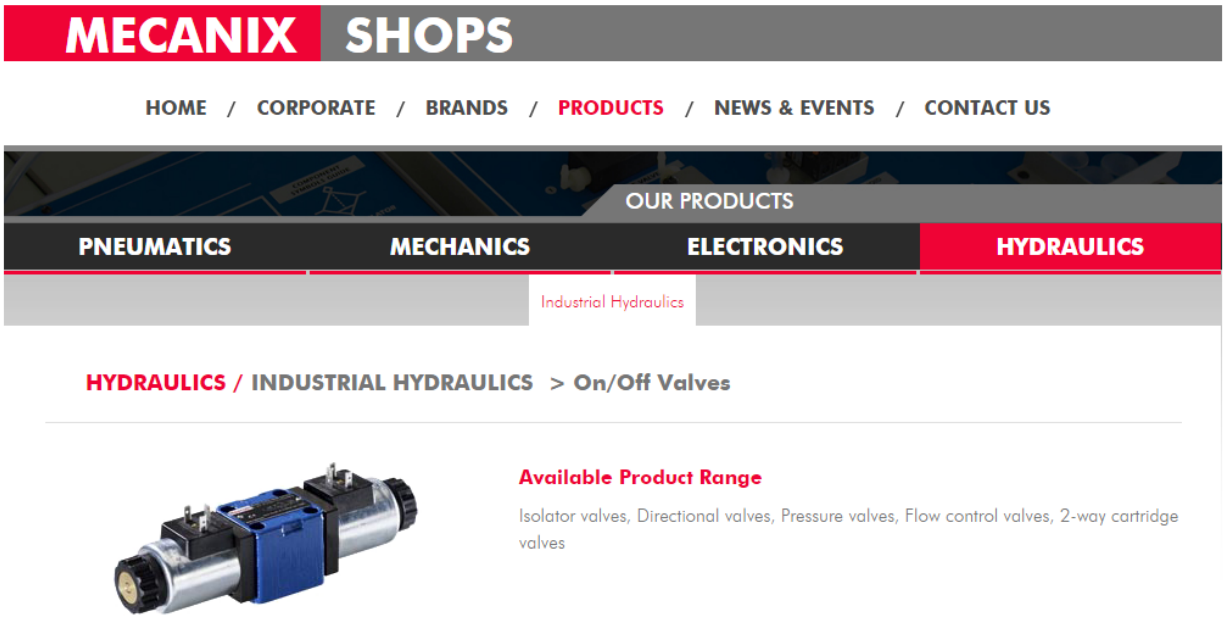
9.1 Mechanical Parts (Valves, Sensors)

Sin El Fil, Horch Tabet
P. : +961-1-486701/2 - +961-1-490754/5
M. : +961-3-783778 / +961-3-763678
F. : +961-1-490929

| +961-76-500880

Mail : P.O. Box 55384 Beirut, Lebanon
Email : sales@mecanixshops.com

9.1.1 Valves



9.1.2 Temperature Sensors

ELECTRONICS / SENSORS > THERMOCOUPLE

Available Product Range

Din Head, Air Probe, MGO, Ceramic, SS316

Product Description:

PT100 EASY-UP

Diameter 6mm, three-wires cable

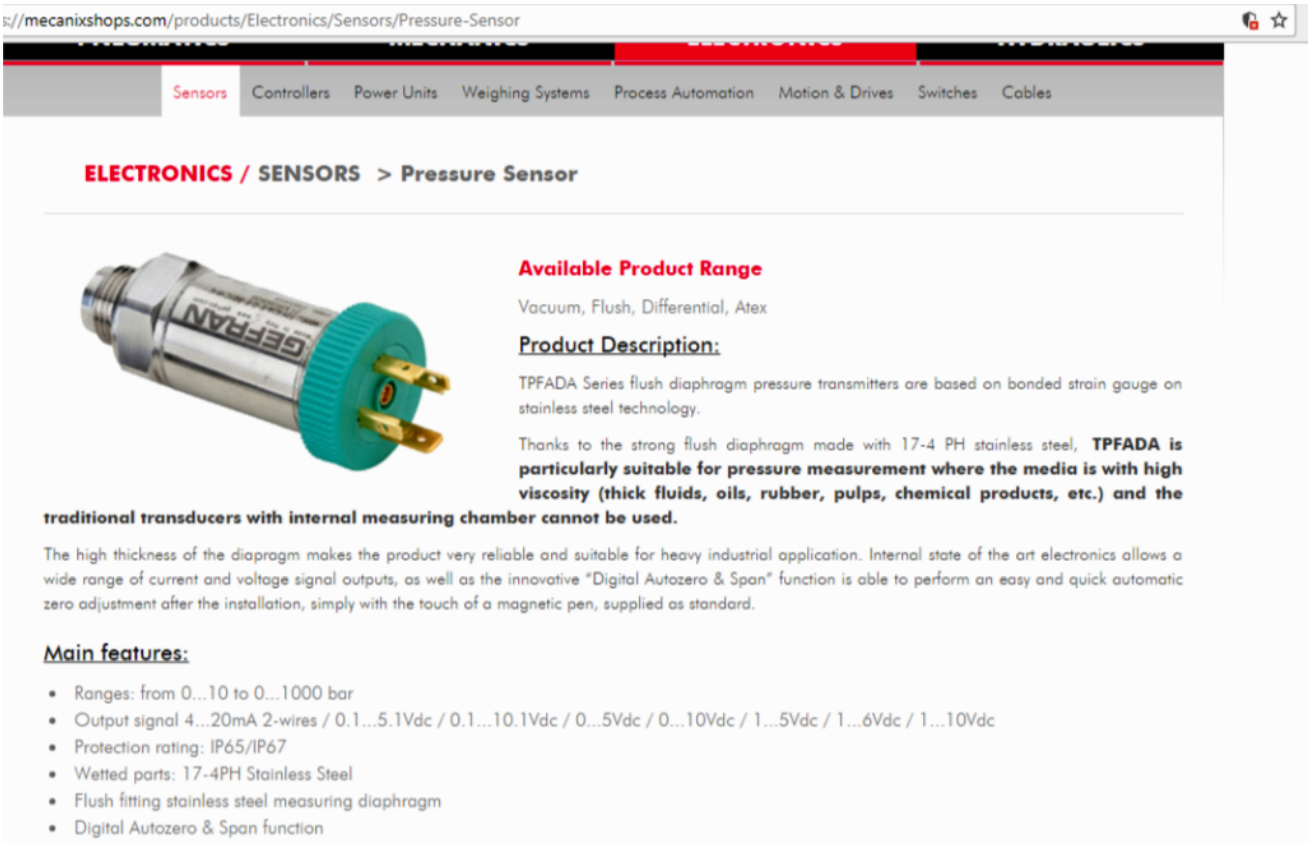
Suppliers

Temperature measurement by resistance thermometers relies on a feature which is common to all conductors and semiconductors: their electrical resistance is subject to variation with changing temperature. PT100 sensor has a resistance of 100 ohms at 0 °C and the coefficient of variation is 0,00385 ohm each °C. The technology to meet requirements of accurate measurement is based on thin platinum layers on a ceramic substrate (thin-film resistors).
A three-wires system is used to compensate the error arising from resistance of connecting cables combined with resistor of the sensor itself.
Easy-Up PT100 are linked to a numeric code (password) to be entered on selected Pixsys controllers in order to set automatically the main operating parameters (type of sensor, measuring range...).

Main features:

- Stem diameter: 6mm
- Immersion material: Steel AISI 304
- Internal insulation: MgO Magnesium oxide > 20mOhm at 25°C (500Vdc)
- Sensor: PT100 class B (+/- 0,3°C up to 25°C)
- GSC cable: Silicon rubber, operating temperature -40 ... 250°C
- TTS cable: Glass fiber, operating temperature -200 ... 500°C
- Sealing: 40mm bend proof spring between cable and stem
- Response time: 6,5 seconds (BS 1904/1984/CEI60751)

9.1.3 Pressure Sensors



9.1.4 Flow Meters

https://mecanixshops.com/products/Electronics/Process-Automation/Flow-Meter

MECANIX SHOPS

HOME / CORPORATE / BRANDS / PRODUCTS / NEWS & EVENTS / COI

OUR PRODUCTS

PNEUMATICS | MECHANICS | ELECTRONICS

Sensors | Controllers | Power Units | Weighing Systems | Process Automation | Motion & Drives | Switch

ELECTRONICS / PROCESS AUTOMATION > Flow Meter

Available Product Range

Magnetic, Ultrasonic, Rotary, Mass

9.1.5 Visualization Software

s://mecanixshops.com/products/Electronics/Process-Automation/Process-visualization-software

OUR PRODUCTS

PNEUMATICS | MECHANICS | ELECTRONICS | HYDRAULICS

Sensors | Controllers | Power Units | Weighing Systems | Process Automation | Motion & Drives | Switches | Cables

ELECTRONICS / PROCESS AUTOMATION > Process Visualization Software

Available Product Range

-Tank configuration-Transmitter configuration-Tankpark visualization-Displaying of measured values-Displaying of limit values-Trend monitoring-Data logging-Database handling-Archiving-Other log functions(alarm)-Remote connection (LAN or Internet)

Product Description:

- -Tank configuration
- -Transmitter configuration
- -Tankpark visualization
- -Displaying of measured values
- -Displaying of limit values
- -Trend monitoring
- -Data logging
- -Database handling
- -Archiving
- -Other log functions (alarms)
- -Remote connection (LAN or Internet)

10 مراجع (References)

[1]http://www.aecenar.com/publications

[2] http://www.aecenar.com/downloads/cat_view/7-megbi-institute

[3] http://www.aecenar.com/downloads/cat_view/3-meae-institute?start=10

[5] NPTEL – Chemical – Chemical Technology II, Joint initiative of IITs and IISc,

[6] https://pypi.python.org/pypi/PyCmdMessenger