

Разработка станка ИТ-42 на основе РС с встроенной платой FPGA Mesa 5I20

г. Ижевск

начато 18 ноября 2010 года

Издание 3-05-2011-01-23

Дата изменения: 2011-01-23

Имеем:

Материнскую плату GIGABIT с процессором Intel P4 2,4GHz

FPGA плату Mesa 5I20

Нужно:

Собрать токарный станок на базе ИТ-42.

Исходные параметры:

Параметр	Ось X	Ось Z
Машинная единица	1 мм	1 мм
Шаг винта ШВП (мм)	4	8
Передаточный коэффициент редуктора двигатель оси/винт ШВП	1	1:2
Число импульсов энкодера на 1 оборот двигателя	2500	2500
Учетверенное число импульсов энкодера на 1 оборот двигателя (для SCALE)	10000	10000
Число импульсов для перемещения на одну машинную единицу	$10000/(4*1)=2500$	$10000/(8*1/2)=2500$
Максимальная скорость движения оси мм/сек (мм/мин)	50 (3000)	50 (3000)
Максимальная скорость вращения двигателя об/мин	$(3000/4*2)=375$	$(3000/8)=375$
Максимальное ускорение движения оси мм/сек. кв.	50	50
Координаты конечных выключателей в отрицательную сторону		
Координаты конечных выключателей в положительную сторону		
Минимальное значение перемещения стола в отрицательную сторону		
Максимальное значение перемещения стола в положительную сторону		

Описание логики работы:

1. Включение станка. Включается только РС. Отдельная кнопка на панели. До ее включения ни одна кнопка не работает. Все выходы платы 5I20 имеют на выходе значение «1». Вся электронная цепь конфигурируется так, что «1» — это безопасное напряжение. Одновременно с включением станка подается напряжение на контакты панели управления. Так как между РС и панелью управления существует оптоэлектрическая изоляция, напряжение на контактах панели управления служит для включения/выключения оптопар, которые далее подключаются изолированными выходами к РС (5I20).
2. С включением РС подается напряжение на плату Mesa 5I20 (она вставлена в PCI слот материнской платы). Система ждет сигнала включения. Этот сигнал подается на пин **iocontrol.0.emc-enable-in**, который связан с *E-STOP*. При отпускании грибка (*E-STOP*), появляется возможность управлять всеми остальными сигналами. Пока *E-STOP* находится в нажатом состоянии, все сигналы игнорируются. После включения *E-STOP* подается напряжение на силовое реле (gr1040). Силовое реле подает напряжение 24 на блок реле. Теперь есть возможность управлять выходными сигналами. Следующий сигнал по цепочке включения подается на пин **halui.machine.on**. Этот сигнал подается кнопкой без фиксированного положения. Кнопка называется *POWER-ON*. Через небольшую задержку после подачи этого сигнала, на выходе пина **halui.machine.is-on** появляется «1». Это говорит о том, что можно включать далее всю остальную силовую часть. Этот сигнал коммутирует цепь питания всех силовых устройств и включает реле P1. Запитывается реле P1 от напряжения контрольной панели. Это единственное реле, которое получает напряжение непосредственно с блока питания. На всех осях на пинах **axis.N.amp-enable-out** так же появляются логические «1». Все оси готовы начать работать.
3. Выключение питания. Кнопкой *POWER-OFF* подается импульс «1» на пин **halui.machine.off**. После поступления этого сигнала на всех пинах **axis.N.amp-enable-out** и **halui.machine.is-on** появляются «0». Вся силовая часть станка отключается.
4. Конечные выключатели и выключатели начального положения. Для безопасности работы устанавливаем конечные выключатели на максимальном и минимальном размерах. Это аварийные выключатели, которые предотвращают физическое повреждение станка. Эти выключатели подсоединяются к следующим пинам:
 - **axis.N.neg-lim-sw-in** — конечный выключатель по минимуму
 - **axis.N.pos-lim-sw-in** — конечный выключатель по максимуму
 При появлении «1» на любом из концевиков любой оси, отключается все силовое питание. На пинах **axis.N.amp-enable-out**, **halui.machine.is-on** устанавливается логический «0». Для разрешения включения силового питания нужно снять логическую «1» с входа концевика.
 - **axis.N.home-sw-in** — вход сигнала *HOME*. Этот сигнал нужен при включении для нахождения начальных координат станка. Используется при процедуре *homing*.
 - Создается кнопка *HOME ALL*. При ее нажатии все оси устанавливаются в «0». Эта кнопка подключена к **halui.home-all**. Но для появления этого пина обязательно нужно создать переменную *HOME_SEQUENCE* в настройках каждой оси. При отсутствии этого параметра пин **halui.home-all** не появится.

5. Кнопка ручного включения подачи охлаждающей жидкости и кнопка ручного выключения подачи охлаждающей жидкости. Подсоединены соответственно к **halui.flood.on** и **halui.flood.off**. Осуществляют подачи на вход РС кратковременного импульса. При включении выход **halui.foood.is-on** устанавливается в «1». При выполнении команды G-кодов M8, M9 выход так же либо устанавливается в «1» либо в «0».
6. Настройка энкодера. Для правильной настройки энкодера нужно рассчитать *INPUT_SCALE*. Формула для расчета *INPUT_SCALE* следующая:
 - $\text{учетверенное_число_импульсов_энкодера_на_1_оборот} \times \text{число_оборотов_для_смещения_на_1_маш_единицу}$.
Для оси X ИТ42 это будет так:
 1. $4 \times 2500 \times 1/4 = 2500$. Это значит 2500 импульсов на 1 оборот, а за один оборот ось проходит 4мм (шаг оси) значит для перемещения на 1 маш. единицу нужно сделать $1/4$ оборота. Вот и получилось 2500 импульсов. Это *INPUT_SCALE* для энкодера.
7. Настройка серводвигателей. Мы используем два серводвигателя SA-030-KM-11-06.0, с крутящим моментом 6Нм. Теперь нам нужно описать взаимодействие компонентов HAL — энкодера, генератора импульсов и PID регулятора. Так как родная конфигурация платы MESA 5120 сконструированная под использования связки PWM + Encoder, и считается что stepgen (генератор импульсов) работает только с шаговым двигателем без обратной связи, нам нужно все делать вручную. Для начала определим режим работы сервопривода. Есть режим по положению и режим по скорости. Нам нужен второй режим — по скорости. Запомним это. Так как при настройке PID модуля и stepgen мы будем задавать именно режим по скорости. Аналогично описываются операции с осью Z. А если осей более, то для каждой оси берется этот пример. При первых испытаниях ошибка слежение (FOLLOW ERROR) составляла до 0,1 мм. Но зато работал INDEX и проблем с HOME уже не должно быть. Дальнейшее совершенствование будет идти по пути настройки привода и подбора оптимальных коэффициентов PID и привода.
8. В качестве дочерней платы мы используем две платы: Mesa 7133TA (4-х канальная серво плата) и Mesa 7137TA (24 канальная плата вход/выход). Для входов и выходов общего назначения мы будем использовать порт P3. Контакты с 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31 используются в качестве входов, а 33,35,37,39,41,43,45,47 в качестве выходов. В HAL они будут идти, как GPIO24-GPIO39 (input), GPIO40-GPIO47 (output). Разводка приведена в таблице.
9. Для отображения статуса сигналов будем использовать руерс панель, что бы оперативно видеть ошибки. Будут созданы HAL пины в качестве светодиодов (led) и на них будут выведены контрольные входные и выходные сигналы.

Приложение №1

Конфигурационный файл станка, необходим для задания основных параметров системе emc2. (Enhanced Machine Controller)

IT42.ini	Описание
<pre> [EMC] MACHINE = IT42-AIKE DEBUG = 0 [DISPLAY] DISPLAY = axis POSITION_OFFSET = RELATIVE POSITION_FEEDBACK = ACTUAL MAX_FEED_OVERRIDE = 2.000000 MAX_SPINDLE_OVERRIDE = 1.000000 MIN_SPINDLE_OVERRIDE = 0.500000 INTRO_GRAPHIC = emc2.gif INTRO_TIME = 5 PROGRAM_PREFIX = /home/operator/emc2/nc_files PYVCP = custompanel.xml INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm LATHE = 1 POSITION_OFFSET = RELATIVE POSITION_FEEDBACK = ACTUAL DEFAULT_LINEAR_VELOCITY = 10.0000 MAX_LINEAR_VELOCITY = 60.000000 MIN_LINEAR_VELOCITY = 1.0000 DEFAULT_ANGULAR_VELOCITY = 0.250000 MAX_ANGULAR_VELOCITY = 1.000000 MIN_ANGULAR_VELOCITY = 0.010000 EDITOR = gedit GEOMETRY = xyz OPEN_FILE = /home/operator/emc2/nc_files/rastochka_9130-4.ngc [FILTER] PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image PROGRAM_EXTENSION = .py Python Script png = image-to-gcode gif = image-to-gcode jpg = image-to-gcode py = python [TASK] TASK = milltask CYCLE_TIME = 0.010 [RS274NGC] PARAMETER_FILE = emc.var [EMCMOT] EMCMOT = motmod COMM_TIMEOUT = 1.0 COMM_WAIT = 0.010 SERVO_PERIOD = 1000000 # [HOSTMOT2] # This is for info only # DRIVER0=hm2_pci # BOARD0=5i20 # CONFIG0="firmware=hm2/5i20/SVST8_4.BIT num_encoders=3 num_pwmgens=1 num_stepgens=2" [HAL] HALUI = halui HALFILE = IT42-AIKE.hal POSTGUI_HALFILE = custom_postgui.hal SHUTDOWN = shutdown.hal [HALUI]</pre>	Имя станка (уникальное). Используем фронтенд AXIS. Координатная система (машинная или относительная) Показывает текущую позицию (заданная или полученная с обратной связи(с мотора)) 200% - максимальное значение FEED_OVERRIDE 100% 0 максимальное значение SPINDLE_OVERRIDE 50% - минимальное значение SPINDLE_OVERRIDE Имя показываемой картинки при запуске. Время показа картинки в секундах. PATH где будут искаться исполняемые программы (G и M). Имя панели создаваемой пользователем. Система будет токарной (LATHE), может быть еще фрезерной (MILL) Линейная скорость по умолчанию в машинных единицах для jog. Максимальная скорость в машинных единицах для jog. Минимальная скорость в машинных единицах для jog. Не используется (нет угловой оси). То же. То же. Имя встроенного редактора. Используемые координаты по порядку до самой старшей (Z, остальные перечисляются по порядку) Какой файл с программой загружать при запуске. Имя основной задачи (разрешено только milltask) Время опроса этой задачи всех устройств. Можно увеличить или уменьшить для точной настройки выполнения заданной задачи. Имя файла, где записываются все параметры, нужные для выполнения программы. Сервопериод в наносекундах. Это длительность основного трида в этой конфигурации. Использовать секцию HAL называемую halui. Нужна для расширения управлением с внешних устройств. Имя основного HAL файла, который нужен для инициализации системы. Файл выполняется только после того, как GUI создаст все HAL pins. Файл, который выполняется при выключения системы.

<pre> [TRAJ] AXES = 3 COORDINATES = X Z LINEAR_UNITS = mm ANGULAR_UNITS = degree CYCLE_TIME = 0.010 DEFAULT_VELOCITY = 100.0 MAX_LINEAR_VELOCITY = 120.0 [EMCIO] EMCIO = io CYCLE_TIME = 0.0100 TOOL_TABLE = tool.tbl ***** # Axis X ***** [AXIS_0] TYPE = LINEAR HOME = 0.0 #experiment 2010-12-17 FERROR = 10.5 #experiment 2010-12-17 MIN_FERROR = 10.5 MAX_VELOCITY = 120.0 MAX_ACCELERATION = 240.0 #Added by hand 2010-11-16 P = 50.0 I = 0.0 D = 0.0 FF0 = 0.0 FF1 = 1.0 FF2 = 0.0 BIAS = 0.0 DEADBAND = 0.001 OUTPUT_SCALE = 500.0 OUTPUT_OFFSET = 0.0 MAX_OUTPUT = 140.0 #??? SCALE INPUT_SCALE = -2500.0 # these are in nanoseconds DIRSETUP = 1000 DIRHOLD = 1000 STEPLEN = 1000 STEPSPACE = 1000 #??? SCALE SCALE = 2500.0 MIN_LIMIT = -338.0 MAX_LIMIT = 2.0 HOME_OFFSET = 0.000000 HOME_SEARCH_VEL = -20.500000 HOME_LATCH_VEL = -1.50000 HOME_FINAL_VEL = 0.000000 HOME_USE_INDEX = YES HOME_SEQUENCE = 1 ***** # Axis Z ***** [AXIS_2] TYPE = LINEAR HOME = 0.0 #experiment 2010-12-17 FERROR = 10.5 </pre>	Число на 1 более самой старшей оси, у нас X (0) Z (2) поэтому 3. Какие оси будет воспринимать g-code интерпретатор. Какие машинные единицы используются (мм или дюймы). Не используется. Начальное значение jog для линейных осей в машинных единицах. Секция работы с внешними устройствами. Время опроса внешних устройств. Имя файла, в котором записывается информация о инструментах. Тип оси — линейная или поворотная. Позиция, на которую будет уходить ось при поиске нуля. Максимальная ошибка слежения. Минимальная ошибка слежения при самой маленькой скорости. Максимальная скорость для этой оси в машинных единицах. Максимальное ускорение для этой оси в машинных единицах на секунду в квадрате. Коэффициент сервоусилителя. Используется при управлении по позиции. Используется при управлении по скорости. Используется при управлении по ускорению. Величина смещения, подаваемая на выход. Зона, в пределах которой не происходит вычисление изменения скорости. Это две связанные величины, OUTPUT_OFFSET (0) вычитается из вычисленного выходного сигнала и делится на OUTPUT_SCALE (1) Максимальное число в вольтах, которое выходит с PID на мотор, в мануале 10В. Число импульсов на для перемещения системы на одну машинную единицу. Данные поступают с энкодера. Время установления импульса направления движения. Минимальная длина импульса направления движения. Минимальная длина импульса STEP Минимальный промежуток между импульсами STEP. Число импульсов для подачи на двигатель, что бы переместить систему по оси на одну машинную единицу. Минимальное значение по этой оси. (softlimit) Максимальное значение по этой оси. Смещение после нахождения нуля по этой оси. Начальная скорость поиска нуля. Скорость после того, как найден переключатель нуля (latch) Скорость от защелки до нуля. Использовать нуль метку энкодера. Последовательность включения оси при нахождении нуля.
--	--

```
#experiment 2010-12-17
MIN_ERROR = 10.5
MAX_VELOCITY = 120.0
MAX_ACCELERATION = 240.0
#Added by hand 2010-11-16
P = 50.0
I = 0.0
D = 0.0
FF0 = 0.0
FF1 = 1.0
FF2 = 0.0
BIAS = 0.0
DEADBAND = 0.001
OUTPUT_SCALE = 250.0
OUTPUT_OFFSET = 0.0
MAX_OUTPUT = 140.0
#change sign 2010-11-16
#???
INPUT_SCALE = 2500.0
```

```
# these are in nanoseconds
DIRSETUP = 1000
DIRHOLD = 1000
STEPLN = 1000
STEPSPACE = 1000
#???
SCALE = 2500.0
MIN_LIMIT = -300.0
MAX_LIMIT = 2.0
HOME_OFFSET = 0.000000
HOME_SEARCH_VEL = -20.500000
HOME_LATCH_VEL = -1.50000
HOME_FINAL_VEL = 0.000000
HOME_USE_INDEX = YES
#HOME_USE_INDEX = NO
HOME_SEQUENCE = 0
```

```
*****
# Spindle
*****
[SPINDLE_9]
MAX_VELOCITY = 16.6666666667
```

```
MAX_ACCELERATION = 20.0
OUTPUT_SCALE = -2000.0
OUTPUT_OFFSET = 0.0
MAX_OUTPUT = 10.0
INPUT_SCALE = 4000.0
```

Максимальная скорость вращения шпинделя в секунду.
Максимальное ускорение шпинделя.
Максимальное число оборотов в минуту

Максимальное выходное напряжение в вольтах.
Число импульсов энкодера для одного оборота.

Приложение №2

Основной файл HAL для управления внешними устройствами станка и взаимодействия его с управляющей программой.

HAL — Hardware Abstraction Layer

IT42.hal	Описание
<pre> loadrt trivkins loadrt [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[TRAJ]AXES loadrt hostmot2 loadrt hm2_pci config="firmware=hm2/5i20/SVST8_4.BIT num_encoders=3 num_pwmgens=1 num_stepgens=2" setp hm2_5i20.0.pwmgen.pwm_frequency 100000 setp hm2_5i20.0.pwmgen.pdm_frequency 100000 setp hm2_5i20.0.watchdog.timeout_ns 10000000 loadrt not count=8 loadrt abs names=abs.spindle loadrt pid num_chan=2 loadrt and2 count=11 loadrt or2 count=1 loadrt oneshot count=2 loadrt timedelay count=2 loadrt mux8 count=1 loadrt flipflop count=4 loadrt minmax count=2 loadrt debounce cfg="6" setp debounce.0.delay 20 addf hm2_5i20.0.read servo-thread addf motion-command-handler servo-thread addf motion-controller servo-thread addf hm2_5i20.0.write servo-thread addf hm2_5i20.0.pet_watchdog servo-thread addf not.0 servo-thread addf not.1 servo-thread addf not.2 servo-thread addf not.3 servo-thread addf not.4 servo-thread addf not.5 servo-thread addf not.6 servo-thread addf not.7 servo-thread addf and2.0 servo-thread addf and2.1 servo-thread addf and2.2 servo-thread addf and2.3 servo-thread addf and2.4 servo-thread addf and2.5 servo-thread addf and2.6 servo-thread addf and2.7 servo-thread addf and2.8 servo-thread addf and2.9 servo-thread addf and2.10 servo-thread addf or2.0 servo-thread addf mux8.0 servo-thread addf oneshot.0 servo-thread addf oneshot.1 servo-thread addf timedelay.0 servo-thread addf timedelay.1 servo-thread addf abs.spindle servo-thread addf flipflop.0 servo-thread addf flipflop.1 servo-thread addf flipflop.2 servo-thread addf flipflop.3 servo-thread addf debounce.0 servo-thread addf minmax.0 servo-thread addf minmax.1 servo-thread addf pid.0.do-pid-calcs servo-thread addf pid.1.do-pid-calcs servo-thread # general signals net strob-man <= halui.mode.is-manual net strob-auto <= halui.mode.is-auto => not.5.in net strob-auto-not <= not.5.out net strob-mdi <= halui.mode.is-mdi net power-on => halui.machine.is-on net lube-is-on => halui.lube.is-on net timer-elapsed => timedelay.0.elapsed net is-running <= halui.program.is-running net is-paused <= halui.program.is-paused net is-idle <= halui.program.is-idle net is-home-x <= halui.joint.0.is-homed => and2.6.in0 net is-home-z <= halui.joint.2.is-homed => and2.6.in1 net is-home-out <= and2.6.out => not.7.in net is-home-out-not <= not.7.out </pre>	

```

# external output signals

# --- ESTOP-OUT ---
net estop-out => not.0.in
net estop-out-not => not.0.out => hm2_5i20.0.gpio.040.out
setp hm2_5i20.0.gpio.040.is_output true
# similar
net estop-out-not => hm2_5i20.0.gpio.042.out
setp hm2_5i20.0.gpio.042.is_output true

# --- ENABLE ---
net enable => not.1.in
net enable-not => not.1.out => hm2_5i20.0.gpio.041.out
setp hm2_5i20.0.gpio.041.is_output true

# external input signals

# --- MIN-X ---
net X.min <= hm2_5i20.0.gpio.024.in
# --- MAX-X ---
net X.max <= hm2_5i20.0.gpio.025.in
# --- HOME-X ---
net X.home <= hm2_5i20.0.gpio.026.in
# --- MIN-Z ---
net Z.min <= hm2_5i20.0.gpio.027.in
# --- MAX-Z ---
net Z.max <= hm2_5i20.0.gpio.028.in
# --- HOME-Z ---
net Z.home <= hm2_5i20.0.gpio.029.in

#####
# START-PROGRAMM #
#####
#signal from BUTTON to MESA card
#test is homed first!!
net is-home-out => and2.7.in0
net start-program-btn <= hm2_5i20.0.gpio.032.in_not => and2.7.in1
#set in TRUE only if all axes homed and pressed button START
net start-program <= and2.7.out
net start-program => and2.5.in0
net strob-auto-not => and2.5.in1
#set auto mode
net set-auto <= and2.5.out
net set-auto => halui.mode.auto
#reset minmax ferror values in both axes
net set-auto => minmax.0.reset
net set-auto => minmax.1.reset
#set oneshot impulse length = 2 sec
setp oneshot.1.width 2
#make impulse lenth 2 sec
net set-auto => oneshot.1.in
net start-program-out => oneshot.1.out
net start-program-out => halui.program.run

#####
# STOP PROGRAMM #
#####
net is-running => and2.8.in0
net stop-program-btn <= hm2_5i20.0.gpio.033.in_not => and2.8.in1
net set-pause <= and2.8.out => halui.program.pause

#det program stop delay 1 sec
setp timedelay.1.on-delay 1
setp timedelay.1.off-delay 0

net is-paused => timedelay.1.in
net stop-program <= timedelay.1.out => halui.program.stop

#####
# LUBE, LUBE-BED ON/OFF #
#####
#set 3 sec delay impulse for first switch-on bed lube
setp oneshot.0.width 3
#switch on bed-lube with machine start
net power-on oneshot.0.in
net delay oneshot.0.out => or2.0.in0
net timer-delay => or2.0.in1
net bed-lube => or2.0.out
net bed-lube => hm2_5i20.0.gpio.044.out
setp hm2_5i20.0.gpio.044.is_output true
setp hm2_5i20.0.gpio.044.invert_output true
#create timer for bed-lube

```

```

#the impulse will be 3 sec
#the delay between impulse will be 60min = 60*60=3600 sec
setp timedelay.0.on-delay 3600
setp timedelay.0.off-delay 3
net lube-is-on => and2.4.in0
net timer-in => and2.4.in1
net timer-in-1 => and2.4.out
net timer-in-1 => timedelay.0.in
net timer-delay => timedelay.0.out
net timer-delay => not.4.in
net timer-in <=& not.4.out

#####
# FLOOD ON/OFF EXTERNAL BUTTON #
#####
net flood-on => halui.flood.on => hm2_5i20.0.gpio.035.in_not
net flood-off => halui.flood.off => hm2_5i20.0.gpio.036.in_not
#out from CNC
net coolant-flood halui.flood.is-on => hm2_5i20.0.gpio.043.out
setp hm2_5i20.0.gpio.043.is_output true
setp hm2_5i20.0.gpio.043.invert_output true

#####
# HOME ALL EXTERNAL BUTTON #
#####
net home-on-btn => and2.1.in0 <=& hm2_5i20.0.gpio.039.in_not
net strob-man => and2.1.in1
net home-strob <=& and2.1.out => halui.home-all

#net is-home-out-not => and2.1.in1
#net home-on <=& and2.1.out => and2.9.in0
#net strob-man => and2.9.in1
#net home-strob <=& and2.9.out => halui.home-all

#####
# SPINDLE ON/OFF EXTERNAL BUTTON #
#####
#START SPINDLE
net spindle-on => and2.2.in0 <=& hm2_5i20.0.gpio.038.in_not
net strob-man => and2.2.in1
net spindle-on-strob <=& and2.2.out => halui.spindle.start

#STOP SPINDLE
net spindle-off => and2.3.in0 <=& hm2_5i20.0.gpio.037.in_not
net strob-man => and2.3.in1
net spindle-off-strob <=& and2.3.out => halui.spindle.stop

# --- SPINDLE-AT-SPEED ---
###net spindle-at-speed <=& hm2_5i20.0.gpio.033.in

#####
# AXIS X
#####
### start add by hand 2010-11-16
setp pid.0.Pgain [AXIS_0]P
setp pid.0.Igain [AXIS_0]I
setp pid.0.Dgain [AXIS_0]D
setp pid.0.bias [AXIS_0]BIAS
setp pid.0.FF0 [AXIS_0]FF0
setp pid.0.FF1 [AXIS_0]FF1
setp pid.0.FF2 [AXIS_0]FF2
setp pid.0.deadband [AXIS_0]DEADBAND
setp pid.0.maxoutput [AXIS_0]MAX_OUTPUT
### end add by hand 2010-11-16

# Step Gen signals/setup

setp hm2_5i20.0.stepgen.00.dirsetup [AXIS_0]DIRSETUP
setp hm2_5i20.0.stepgen.00.dirhold [AXIS_0]DIRHOLD
setp hm2_5i20.0.stepgen.00.steplen [AXIS_0]STEPLEN
setp hm2_5i20.0.stepgen.00.stepspace [AXIS_0]STEPSPACE
setp hm2_5i20.0.stepgen.00.position-scale [AXIS_0]SCALE
setp hm2_5i20.0.stepgen.00.maxaccel 0
setp hm2_5i20.0.stepgen.00.maxvel 0
setp hm2_5i20.0.stepgen.00.step_type 0
setp hm2_5i20.0.stepgen.00.control-type 1

# ---Encoder feedback signals/setup---

setp hm2_5i20.0.encoder.00.counter-mode 0
setp hm2_5i20.0.encoder.00.filter 1
setp hm2_5i20.0.encoder.00.index-invert 0
setp hm2_5i20.0.encoder.00.index-mask 0
setp hm2_5i20.0.encoder.00.index-mask-invert 0

```

```

setp hm2_5i20.0.encoder.00.scale [AXIS_0]INPUT_SCALE

net X.pos-fb axis.0.motor-pos-fb <= hm2_5i20.0.encoder.00.position => pid.0.feedback
net X.pos-cmd axis.0.motor-pos-cmd => pid.0.command
net X.enable axis.0.amp-enable-out => hm2_5i20.0.stepgen.00.enable => pid.0.enable
net X.pos-out pid.0.output => hm2_5i20.0.stepgen.00.velocity-cmd
net X.index => hm2_5i20.0.encoder.00.index-enable <= axis.0.index-enable

# ---setup home / limit switch signals---

net X.home => debounce.0.4.in
net X.home-filtred => debounce.0.4.out axis.0.home-sw-in
net X.min => debounce.0.0.in
net X.min-filtred => debounce.0.0.out axis.0.neg-lim-sw-in
net X.max => debounce.0.1.in
net X.max-filtred => debounce.0.1.out axis.0.pos-lim-sw-in
net X.ferror => axis.0.f-error
net X.ferror => minmax.0.in

#####
# AXIS Z
#####
### start add by hand 2010-11-16
setp pid.1.Pgain [AXIS_0]P
setp pid.1.Igain [AXIS_0]I
setp pid.1.Dgain [AXIS_0]D
setp pid.1.bias [AXIS_0]BIAS
setp pid.1.FF0 [AXIS_0]FF0
setp pid.1.FF1 [AXIS_0]FF1
setp pid.1.FF2 [AXIS_0]FF2
setp pid.1.deadband [AXIS_0]DEADBAND
setp pid.1.maxoutput [AXIS_0]MAX_OUTPUT
### end add by hand 2010-11-16

# Step Gen signals/setup

setp hm2_5i20.0.stepgen.01.dirsetup [AXIS_2]DIRSETUP
setp hm2_5i20.0.stepgen.01.dirhold [AXIS_2]DIRHOLD
setp hm2_5i20.0.stepgen.01.steplen [AXIS_2]STEPLEN
setp hm2_5i20.0.stepgen.01.stepspace [AXIS_2]STEPSPACE
setp hm2_5i20.0.stepgen.01.position-scale [AXIS_2]SCALE
setp hm2_5i20.0.stepgen.01.maxaccel 0
setp hm2_5i20.0.stepgen.01.maxvel 0
setp hm2_5i20.0.stepgen.01.step_type 0
setp hm2_5i20.0.stepgen.01.control-type 1

# ---Encoder feedback signals/setup---

setp hm2_5i20.0.encoder.01.counter-mode 0
setp hm2_5i20.0.encoder.01.filter 1
setp hm2_5i20.0.encoder.01.index-invert 0
setp hm2_5i20.0.encoder.01.index-mask 0
setp hm2_5i20.0.encoder.01.index-mask-invert 0
setp hm2_5i20.0.encoder.01.scale [AXIS_2]INPUT_SCALE

net Z.pos-fb axis.2.motor-pos-fb <= hm2_5i20.0.encoder.01.position => pid.1.feedback
net Z.pos-cmd axis.2.motor-pos-cmd => pid.1.command
net Z.enable axis.2.amp-enable-out => hm2_5i20.0.stepgen.01.enable => pid.1.enable
net Z.pos-out pid.1.output => hm2_5i20.0.stepgen.01.velocity-cmd
net Z.index => hm2_5i20.0.encoder.01.index-enable <= axis.2.index-enable

# ---setup home / limit switch signals---

net Z.home => debounce.0.5.in
net Z.home-filtred => debounce.0.5.out axis.2.home-sw-in
net Z.min => debounce.0.2.in
net Z.min-filtred => debounce.0.2.out axis.2.neg-lim-sw-in
net Z.max => debounce.0.3.in
net Z.max-filtred => debounce.0.3.out axis.2.pos-lim-sw-in
net Z.ferror => axis.2.f-error
net Z.ferror => minmax.1.in

#####
# SPINDLE S
#####

# ---PWM Generator signals/setup---

setp hm2_5i20.0.pwmgen.00.output-type 1
setp hm2_5i20.0.pwmgen.00.scale [SPINDLE_9]OUTPUT_SCALE

net spindle-enable => hm2_5i20.0.pwmgen.00.enable
net spindle-vel-cmd => abs.spindle.in
net absolute-spindle-vel abs.spindle.out => hm2_5i20.0.pwmgen.00.value

```

```

net spindle-ccw => not.2.in
net spindle-enable => not.3.in
net spindle-cw-n => not.2.out => hm2_5i20.0.gpio.045.out
setp hm2_5i20.0.gpio.045.is_output true
net spindle-enable-n => not.3.out => hm2_5i20.0.gpio.047.out
setp hm2_5i20.0.gpio.047.is_output true

# ---Encoder feedback signals/setup---

setp hm2_5i20.0.encoder.02.counter-mode 0
setp hm2_5i20.0.encoder.02.filter 1
setp hm2_5i20.0.encoder.02.index-invert 0
setp hm2_5i20.0.encoder.02.index-mask 0
setp hm2_5i20.0.encoder.02.index-mask-invert 0
setp hm2_5i20.0.encoder.02.scale [SPINDLE_9]INPUT_SCALE

net spindle-revs <= hm2_5i20.0.encoder.02.position
net spindle-vel-fb <= hm2_5i20.0.encoder.02.velocity
net spindle-index-enable <=> hm2_5i20.0.encoder.02.index-enable

# ---setup spindle control signals---

net spindle-vel-cmd-rps <= motion.spindle-speed-out-rps
net spindle-vel-cmd <= motion.spindle-speed-out
net spindle-enable <= motion.spindle-on
net spindle-cw <= motion.spindle-forward
net spindle-ccw <= motion.spindle-reverse
net spindle-brake <= motion.spindle-brake
net spindle-revs => motion.spindle-revs
net spindle-at-speed => motion.spindle-at-speed
net spindle-vel-fb => motion.spindle-speed-in
net spindle-index-enable <=> motion.spindle-index-enable
##### POSITIVELY
#if false, then input encoder
#if true - no input encoder
sets spindle-at-speed true
#####
# connect miscellaneous signals
#####

# ---probe signal---

net probe-in => motion.probe-input

# ---estop signals---

#logical AND
net estop-in1 and2.0.in0 <= iocontrol.0.user-enable-out
#if button pressed, then "0" everething
net estop-in2 and2.0.in1 <= hm2_5i20.0.gpio.034.in_not
net estop-out and2.0.out => iocontrol.0.emc-enable-in
net enable => motion.motion-enabled

# ---toolchange signals for custom tool changer---

#net tool-number <= iocontrol.0.tool-prep-number
#net tool-change-request <= iocontrol.0.tool-change
#net tool-change-confirmed => iocontrol.0.tool-changed
#net tool-prepare-request <= iocontrol.0.tool-prepare
#net tool-prepare-confirmed => iocontrol.0.tool-prepared

net tool-number <= iocontrol.0.tool-prep-number
net tool-change-loopback iocontrol.0.tool-change => iocontrol.0.tool-changed
net tool-prepare-loopback iocontrol.0.tool-prepare => iocontrol.0.tool-prepared

```

Приложение №3

Таблица соответствия контактов платы MESA 5I20 с внешними устройствами и сигналами.

№ п/п	Наименование кнопки (контакта, реле)	Пин подключения HAL	Направление (со стороны PC)	Использовать	Номер контакта HAL (плата)	Описание сигнала
1	X min lim (конечник)	axis.0.neg-lim-sw-in	IN	Да	Gpio.024(P3-1)	Конечник по минимуму X
2	X max lim (конечник)	axis.0.pos-lim-sw-in	IN	Да	Gpio.025(P3-3)	Конечник по максимуму X
3	X home (конечник)	axis.0.home-sw-in	IN	Да	Gpio.026(P3-5)	Указатель нулевой отметки по X
4	Z min lim (конечник)	axis.2.neg-lim-sw-in	IN	Да	Gpio.027(P3-7)	Конечник по минимуму Z
5	Z max lim (конечник)	axis.2.pos-lim-sw-in	IN	Да	Gpio.028(P3-9)	Конечник по максимуму Z
6	Z home (конечник)	axis.2.home-sw-in	IN	Да	Gpio.029(P3-11)	Указатель нулевой отметки по Z
7	X ready	Axis.0.amp-fault-in	IN	Да	Gpio.030(P-13)	Сигнал готовности с X преобразователя
8	Z ready	Axis.2.amo-fault-in	IN	Да	Gpio.031(P-15)	Сигнал готовности с Z преобразователя
9	START	Halui.program.run	IN	Да	Gpio.032(P-17)	Пуск выполняемой программы
10	STOP	Halui.program.stop	IN	Да	Gpio.033(P-19)	Остановка выполняемой команды
11	E-STOP (кнопка с фиксацией)	iocontrol.0.emc-enable-in	IN	Да	Gpio.034(P3-21)	Аварийная остановка станка (ESTOP). Кнопка с фиксацией. При приходе TRUE сигнал активируется и сбрасывает E-STOP.
12	Coolant On (кнопка)	halui.flood.on	IN	Да	Gpio.035(P3-23)	Сигнал на включение охлаждения. Обрабатывается вручную. Кнопка без фиксации.
13	Coolant Off (кнопка)	halui.flood.off	IN	Да	Gpio.036(P3-25)	Сигнал на выключение охлаждения, обрабатывается вручную. Кнопка без фиксации.
14	Spindle Stop	halui.spindle.stop	IN	Да	Gpio.037(P3-27)	Ручное выключение шпинделя. Кнопка без фиксации.
15	Spindle Start	halui.spindle.start	IN	Да	Gpio.038(P3-29)	Ручное включение шпинделя. Кнопка без фиксации.
16	HOME ALL (кнопка)	halui.home-all	IN	Да	Gpio.039(P3-31)	Запрос на установку всех осей в «0». Требуется задать HOME_SEQUENCE в ini файле. Кнопка без фиксации.
17	E-STOP-OUT-NOT	Iocontrol.0.user-enable-out && hm2_5i20.0.gpio.034.in_not	OUT	Да	Gpio.040(P3-33)	Включает реле силового питания и подает питание на сервопривода. При нажатой E-STOP выход = TRUE, при отжатой = FALSE. Инверсный сигнал от ESTOP-OUT
18	AXIS enable (реле)	motion.motion-enabled (инверсный)	OUT	Да	Gpio.041(P3-35)	Подает сигнал разрешения включения на все сервопривода. Активный 0.
19			OUT	Нет	Gpio.042(P3-37)	
20	Coolant IsOn (реле)	halui.flood.is-on	OUT	Да	Gpio.043(P3-39)	Сигнал на реле для включения струи охлаждения. Обрабатывается как M8/M9 так и в ручную.
21	Lube IsOn (реле)	halui.lube.is-on && bed-lube	OUT	Да	Gpio.044(P3-41)	Выходной сигнал на реле включает насос смазки направляющих. Работает по тайм-ауту. Один раз на 3 секунды в 1 час.
22	spindle-ccw-not	spindle-ccw-not	OUT	Да	Gpio.045(P3-43)	Задание направления включения шпинделя.
23			OUT	Нет	Gpio.046(P3-45)	
24	spindle-enable	spindle-enable	OUT	Да	Gpio.047(P3-47)	Разрешение включение привода шпинделя.
25	Энкодер В (Z)		IN	Да	Gpio.000(P2-1)	Энкодер сервопривода оси Z (В)
26	Энкодер А (Z)		IN	Да	Gpio.001(P2-3)	Энкодер сервопривода оси Z (А)
27	ЭнкодерВ (X)		IN	Да	Gpio.002(P2-5)	Энкодер сервопривода оси X (В)
28	Энкодер А (X)		IN	Да	Gpio.003(P2-7)	Энкодер сервопривода оси X (А)
29	Энкодер Index (Z)		IN	Да	Gpio.004(P2-9)	Энкодер сервопривода оси Z (Index)
30	Энкодер Index (X)		IN	Да	Gpio.005(P2-11)	Энкодер сервопривода оси X (Index)
31				Нет	Gpio.006(P2-13)	

32	PWMGen Impulse		OUT	Да	Gpio.007(P2-15)	Управление скоростью вращения шпинделя.
33				Нет	Gpio.008(P2-17)	
34	PWMGen Dir		OUT	Да	Gpio.009(P2-19)	Направление вращения шпинделя, прямое или обратное.
35				Нет	Gpio.010(P2-21)	
36	PWMGen Enable		OUT	Да	Gpio.011(P2-23)	Разрешение включения шпинделя.
37				Нет	Gpio.012	
38				Нет	Gpio.013	
39				Нет	Gpio.014	
40				Нет	Gpio.015	
41				Нет	Gpio.016	
42				Нет	Gpio.017	
43				Нет	Gpio.018	
44				Нет	Gpio.019	
45				Нет	Gpio.020	
46				Нет	Gpio.021	
47				Нет	Gpio.022	
48				Нет	Gpio.023	
49	Step X (опто соединение)	hm2_5120.0.stepgen.00.st ep	OUT	Да	Gpio.048(P4-1)	Сигнал на сервопривод X шаг (STEP X)
50	Dir X (опто соединение)	hm2_5120.0.stepgen.00.di r	OUT	Да	Gpio.049(P4-3)	Сигнал на сервопривод X направление (DIR X)
51				Нет	Gpio.050	
52				Нет	Gpio.051	
53				Нет	Gpio.052	
54				Нет	Gpio.053	
55	Step Z (опто соединение)	hm2_5120.0.stepgen.01.st ep	OUT	Да	Gpio.054(P4-13)	Сигнал на сервопривод Z шаг (STEP Z)
56	Dir Z (опто соединение)	hm2_5120.0.stepgen.01.di r	OUT	Да	Gpio.055(P4-15)	Сигнал на сервопривод Z направление (DIR Z)
57				Нет	Gpio.056	
58				Нет	Gpio.057	
59				Нет	Gpio.058	
60				Нет	Gpio.059	
61				Нет	Gpio.060	
62				Нет	Gpio.061	
63				Нет	Gpio.062	
64				Нет	Gpio.063	
65				Нет	Gpio.064	
66				Нет	Gpio.065	
67				Нет	Gpio.066	
68				Нет	Gpio.067	
69				Нет	Gpio.068	
70				Нет	Gpio.069	
71				Нет	Gpio.070	
72				Нет	Gpio.071	

Приложение №4

Список дополнительных сигналов, не используемых в настоящий момент.

№ п/п	Наименование кнопки (контакта, реле)	Пин подключения HAL	Направление (со стороны PC)	Использовать	Номер контакта HAL (плата)	Описание сигнала
	Power-On (кнопка)	halui.machine.on	IN	Нет		Включение силового питания (Power On), кнопка без фиксации
	Power-Off (кнопка)	halui.machine.off	IN	Нет		Выключение силового питания (Power Off), кнопка без фиксации.
	Power Enable (реле)	halui.machine.is-on	OUT	Нет		Выдает сигнал на включение питания коммутации силовых устройств.
	Spindle ready		IN	Нет		Приходит со шпинделя о готовности к работе.
	Spindle Check		IN	Нет		Приходит со шпинделя N=0 сигнал об остановке шпинделя.
	X ready	axis.0.amp-fault-in	IN	Нет		Приходит с сервопривода оси X о готовности к работе, TRUE — привод не готов. Для работы сигнал должен быть FALSE.
	Z ready	axis.2.amp-fault-in	IN	Нет		Приходит с сервопривода оси Z о готовности к работе, TRUE — привод не готов. Для работы сигнал должен быть FALSE.
	Spindle increase	halui.spindle.increase	IN	Нет		Ручное увеличение оборотов шпинделя. Кнопка без фиксации.
	Spindle decrease	halui.spindle.decrease	IN	Нет		Ручное уменьшение оборотов шпинделя. Кнопка без фиксации.
	AUTO MODE (кнопка)	halui.mode.auto	IN	Нет		Включение автоматического режима. Галетный переключатель.
	MANUAL MODE	halui.mode.manual	IN	Нет		Включение ручного режима. Галетный переключатель.
	MDI MODE	halui.mode.mdi	IN	Нет		Включение MDI режима. Галетный переключатель.
	RUN PROGRAMM (кнопка)	halui.program.run	IN	Нет		Старт программы в автоматическом режиме. Кнопка без фиксации.
	PAUSE PROGRAMM	halui.program.pause	IN	Нет		Пауза при выполнении программы. Кнопка без фиксации.
	RESUME PROGRAMM	Halui.program.resume	IN	Нет		Продолжение выполнения программы. Кнопка без фиксации.
	STOP PROGRAMM (кнопка)	halui.abort	IN	Нет		Остановка программы (только через нажатие PAUSE). Кнопка без фиксации.
	JOG X+ (кнопка)	halui.jog.0.plus	IN	Нет		
	JOG Z+ (кнопка)	halui.jog.2.plus	IN	Нет		
	JOG X- (кнопка)	halui.jog.0.minus	IN	Нет		
	JOG Z- (кнопка)	halui.jog.2.minus	IN	Нет		
	JOG SCALE 1 (трехсекционный переключатель)	jog-scale-1	IN	Нет		
	JOG SCALE 2 (трехсекционный переключатель)	jog-scale-2	IN	Нет		
	JOG SCALE 3 (трехсекционный переключатель)	jog-scale-3	IN	Нет		

ИТОГО:

Входных сигналов — $6+16+0 = 22$ Выходных сигналов — $3+5+4 = 12$

Наличие по платам 7I37TA (2 шт.) и 7I33TA (1 шт.)

Входных сигналов — 32

Выходных сигналов — 16

№ п/п	Наименование индикатора	Пин подключения HAL	Направление (со стороны РС)	Использовать	Пин подключения РYVCP	Описание сигнала
1	X min fault (светодиод)	axis.0.neg-hard-limit	OUT	Да		Достижение минимального предела по оси X.
2	X max fault (светодиод)	axis.0.pos-hard-limit	OUT	Да		Достижение максимального предела по оси X.
3	Z min fault (светодиод)	axis.2.neg-hard-limit	OUT	Да		Достижение минимального предела по оси Z.
4	Z max fault (светодиод)	axis.2.pos-hard-limit	OUT	Да		Достижение максимального предела по оси Z.