پروژه پایانی

محمدمهدی تیموریان و فاطمه جعفری

چکیده

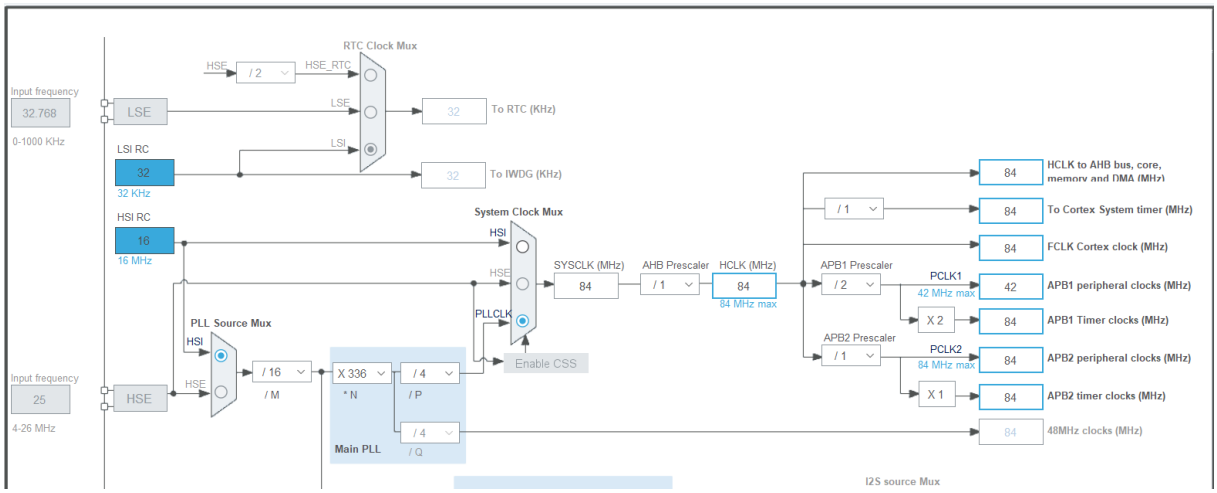
برنامه طراحی شده با تولید موج PWM، انواع موج های مختلف مانند سینوسی، مثلثی، مربعی و غیره را تولید میکند. همچنین میتوان نوع موج، فرکانس و دامنه را تغییر داد. ورودی ها با استفاده از مقاومت متغیر و سویچ به میکروکنترلر داده و سپس با وقفه ها مدیریت و پردازش میشوند. همچنین خروجی موج PWM با عبور از ۲ فیلتر سلسله‌ای به موج آنالوگ تبدیل میشود.

تنظیمات کلاک

کلاک میکروکنترلر به صورت پیش‌فرض برابر با ۱۶Mhz می باشد. برای فراهم کردن فرکانس های بالاتر در موج خروجی و همچنین کیفیت بیشتر موج خروجی کلاک را با استفاده از تنظیمات زیر به ۸۴Mhz تغییر میدهیم:

با انتخاب HSI به عنوان منبع PLL و انتخاب PLL به عنوان کلاک سیستم، میتوان مقدار کلاک را به ۱۶۸Mhz برسانیم. با توجه به محدودیت های STM۳۲F۴۰۱RE ماکسیمم مقدار برای تایمر ها ۸۴Mhz می باشد. در نتیجه با تنظیم پارامترهای PLL و تنظیم Prescaler باس های ادوات به این مقدار دست می‌یابیم.

بخش	پارامتر	مقدار
PLL	M	۱۶
	N	۳۳۶
	P	۴
	Q	۴
باس AHB	AHB Prescaler	۱
باس ادوات APB۱	APB۱ Prescaler	۲
باس ادوات APB۲	APB۲ Prescaler	۱



تنظیمات PWM

با استفاده از تایمر ۵ موج PWM تولید میشود. تایمر ۲ با ایجاد وقفه در بازه هایی متناسب با فرکانس موج PWM، مقدار duty cycle را از جدول موج مورد نظر خوانده و با در نظر گرفتن دامنه موج، بر روی تایمر مربوط به PWM تنظیم میکند. مقدار ARR همیشه ثابت می ماند و برابر با ۴۰۹۵ است. این مقدار در سیستم باینری ۱۲ بیت نیاز دارد که به میزان resolution موج خروجی اشاره میکند. همچنین مقدار PSC تایمر تولید کننده PWM، متناسب با فرکانس تنظیم شده توسط کاربر، تغییر میکند.

ویژگی	مقدار
دامنه موج	[۴,۱۰]
فرکانس موج	[۱,۲۰K]
نمونه سمپل به ازای هر موج	۱۲۸
رزولوشن موج خروجی	۱۲

جداول موج‌ها

هر موج یک جدول از پیش محاسبه شده دارد که مقادیر duty cycle موج خروجی از این جدوال خوانده میشود. تعداد ۱۲۸ نمونه در هر جدول وجود دارد که متناسب با رزولوشن درنظر گرفته شده است و کیفیت خوب و مناسبی را در موج خروجی ارائه میدهد.

تغییر دامنه موج

مقدار duty cycle موج خروجی از جدول متناسب با انتخاب کاربر، خوانده میشود. سپس با بررسی مقدار دامنه موج مقدار duty cycle را به یک دامنه کوچکتر یا بزرگتر مقایس دهی میکند.

تغییر فرکانس موج

در صورت تغییر فرکانس توسط کاربر، به صورت لحظه ای مقدار جدید PSC متناسب با کلاک تایمر، رزولوشن و فرکانس تنظیم شده توسط کاربر محاسبه و بر روی تایمر های ۵ و ۲ تنظیم میشود.

محاسبات مربوط به PWM

نحوه محاسبه amplitude و frequency در بخش خواندن ورودی آنالوگ توضیح داده شده است.

$$PSC = \frac{TIM\ CLK}{Frequency * 2^{Resolution}}; resolution = 12$$

$$Wave\ Scale = \frac{2^{Resolution} * Amplitude}{Max\ Amplitude} - 2048\ ; resolution = 12$$

$$Duty\ Cycle = \frac{LUT_i * 2 * Wave\ Scale}{2^{Resolution}} + 2048 - Wave\ Scale\ ; resolution = 12$$

فیلتر موج PWM

برای گرفتن موج آنالوگ بهتر از دو فیلتر به صورت سلسله‌ای استفاده شده است. ابتدا با استفاده از یک LCF نوع T یکبار نویزهای موج گرفته میشود. خروجی موج از فیلتر اول به فیلتر دوم داده میشود تا کیفیت موج بالا رود و نویز آن کمتر شود. فیلتر نوع دوم یک LCF از نوع L می‌باشد.

با استفاده از معادلات مدارهای خازن القاگری و همچنین آزمون و خطا و مشاهده خروجی، بهترین مقادیر برای این فیلتر ها انتخاب شده که شرح آن در جدول زیر آمده است.

فیلتر	پارامتر	مقدار
LCF-T	هر دو القاگر	1mH
LCF-T	خازن	5u
LCF-L	القاگر	1mH
LCF-L	خازن	10u

خواندن ورودی آنالوگ

رزولوشن انتخابی برای 12 می باشد. علت این است که بتوانیم رنج کامل ولتاژ ورودی را دریافت کنیم و مقدار آن را به بازه 0 تا 100 مقایس دهیم. برای دریافت پارامترهای ورودی که شامل دامنه و فرکانس شکل موج خروجی است، از ادوات ADC1 و TIM3 و از دو پین PA0, PA1 استفاده شده است. که این دو پین به دو کانال (regular channel) از ADC1 پردازنده متصل می شود (SQ1, SQ2 در ADC_SQR3) .هر کدام از این کانال ها در مود sample time بیشینه (480 cycles) هستند. برای TIM3 نیز update interrupt enable فعال شده است و با Prescaler = 1000-1 و auto reload register = 750-1 به وسیله یک interrupt ، ADC را به وسیله پین SWSTART روشن می کند و در interrupt مربوط به ADC بسته به این که خروجی ADC از کدام کانال است، مقدار frequency و یا amplitude در صورت تغییر ست می شوند. و پس ست شدن، با توجه به

درصد مقدار خوانده شده از ADC که همان میزان مقاومت متغیر ها را نشان می دهد به صورت پله ای مقادیر مناسب برای فرکانس و دامنه حساب می شود و با ست کردن یک flag که در صورت تغییر هر یک از ورودی ها (فرکانس، دامنه و نوع موج) ست می شود، و همزمان با تولید شکل موج روی نمایشگر LCD نمایش داده می شود. محاسبه و میزان هر پرش برای فرکانس و دامنه به صورت زیر است :

$$Frequency\ Percentage = round\left(\frac{Input\ Frequency\ Portion}{Input\ Resolution}\right) * 100$$

$$Amplitude\ Percentage = round\left(\frac{Input\ Amplitude\ Portion}{Input\ Resolution}\right) * 100$$

دامنه معادل	درصد مقاومت متغیر دامنه	Step size	فرکانس معادل(Hz)	درصد مقاومت متغیر فرکانس
4.0	0 to 14	1	1 to 10	0 to 10
5.0	15 to 29	10	20 to 100	11 to 20
6.0	30 to 44	50	150 to 1000	21 to 38
7.0	45 to 59	100	1100 to 5000	39 to 78
8.0	60 to 74	250	5250 to 10000	79 to 98
9.0	75 to 89	-	15000	99
10.0	90 to 100	-	20000	100

ترمینال مجازی و دیباگ

برای دیباگ کردن و فهمیدن مقدار دیجیتال شده (خروجی ADC) از USART1 استفاده شده است که صرفا از این دیوایس برای نوشتن و چک کردن پارامترها استفاده می شود. برای این منظور پین PA9 برای ارسال (پین PA10 برای دریافت که مورد استفاده نیست) در نظر گرفته شده است. در خارج از پردازنده این پین به یک VIRTUAL TERMINAL که برای نمایش به کار می رود متصل است. برای اینکار نیاز است که PA9 را در مود AF قرار دهیم. سپس از تابع usart_write برای نوشتن داده و ارسال آن به VT (ترمینال) استفاده می شود. از

این دیوایس برای دیباگ کردن، در بخش ADC_IRQHandler برای زمانی که مقدار پارامتر های ورودی (فرکانس و دامنه موج) تغییر می کنند استفاده می شود که مقادیر زیر، در صورت تغییر، در فرمت ذکر شده نمایش داده می شوند:

- دامنه ورودی که از ADC1 خوانده شده و به درصد تبدیل شده است.
- فرکانس ورودی که از ADC1 خوانده شده و به درصد تبدیل شده است.
- Prescaler مورد نظر برای تولید شکل موج که به صورت CLK/(frequency*RES) محاسبه می شود (مقدار CLK, RES ثابت های مورد نظر برای فرکانس کلاک و رزولوشن و frequency مقدار فرکانس ورودی است.) به همراه فرکانس تولید کننده آن (فرکانس ورودی).

نمایشگر

برای نمایش مقادیر اولیه که شامل فرکانس، دامنه و نوع موج ورودی است از LCD استفاده شده است که پین های آن به پورت C متصل شده اند (پین های تنظیمات روی PC0 تا PC2 و پین های دیتا روی PC3 تا PC10 هستند). برای LCD باید در ابتدای کار باید initialize کنیم و سپس با استفاده از دستورات زیر، کار مورد نظر را انجام دهیم:

0x38	function set: 8-bit, 2-line, 5x7 font
0x06	move cursor right after each char
0x0C	turn on display, cursor off, no cursor blinking
0x01	clear screen, move cursor to home
0xC0	move cursor to next line
0x80	force cursor to beginning to 1st line

تمامی دستورات بالا، باید به وسیله تابع display_command فراخوانی شوند. در این تابع در هر مرحله پس از تنظیمات جدول زیر، command مورد نظر روی خروجی قرار می گیرد (به وسیله باس داده پورت C) و سپس بیت enable = 0 می شود:

E = 1	enable
RW = 0	data write(mp->lcd)
DI = 0	instruction input

برای نوشتن یک کاراکتر رو lcd کافیست در هر مرحله پس از تنظیمات جدول زیر، کاراکتر مورد نظر روی خروجی قرار بگیرد (به وسیله باس داده پورت C) و سپس بیت enable = 0 می شود:

E = 1	enable
RW = 0	data write(mp->lcd)
DI = 1	data input

برای نوشتن یک string کافیست کاراکتر به کاراکتر رشته مورد نظر را روی lcd نمایش دهیم.

ورودی نوع موج

از آن جایی که 8 نوع موج داریم به 8 سوییچ نیاز داریم که از یک DSWITCH 8 تایی استفاده می کنیم و پین های آن را به PB5 تا PB13 متصل می کنیم. همه این پین ها امکان تغییر در طول برنامه را دارد (تضمین می شود هیچ 2 پینی با هم 1 نمی شوند). به همین علت این پین ها را به external interrupt های 5 تا 13 متصل میکنیم تا پس از هر تغییر interrupt مناسب داده شود. پس از هربار دریافت interrupt نوع موج دریافتی ست می شود و interrupt مربوطه mask و در صورت تغییر نوع موج دریافتی، مقادیر ورودی که روی lcd نمایش داده می شدند آپدیت می شود.

رابطه بین پین ها، نوع موج و interrupt مربوطه به صورت زیر است:

نوع موج	پین متناظر	Interrupt متناظر
Sine Wave	PB5	EXTI5
Square Wave	PB6	EXTI6
Triangle Wave	PB7	EXTI7
Sawtooth Wave	PB8	EXTI8

Stairs Wave	PB9	EXTI9
Rectified Sine Wave	PB10	EXTI10
Segmented Sine Wave	PB12	EXTI12
Singe Modulation Wave	PB13	EXTI13

ورودی خروجی‌ها

پین	نوع	توضیحات
PA0	ورودی	آنالوگ فرکانس موج
PA1	ورودی	آنالوگ دامنه موج
PA2	خروجی	موج PWM
PA9	خروجی	ارتباط سریال با ترمینال مجازی
PB0	خروجی	ال ای دی کمکی
PB5-13	ورودی	سویچ‌ها برای انتخاب موج
PC0-10	خروجی	ارتباط با نمایشگر

ادوات استفاده شده

ادوات	توضیحات
TIM2	استفاده برای نوشتن مقدار خوانده شده از جدول موج درون PSC
TIM3	استفاده برای نمونه برداری از دو ورودی فرکانس و دامنه موج
TIM5	استفاده برای تولید موج PWM
ADC1	استفاده برای خواندن مقادیر آنالوگ دو ورودی فرکانس و دامنه موج
USART1	استفاده برای دیباگ و لاگ انداخت در ترمینال مجازی

بهینه‌سازی

ابتدا برای افزایش سرعت عمل از جداول از پیش محاسبه شده برای موج ها استفاده شده است. این عمل کمک میکند بدون وقفه duty cycle عوض شود و شکل موج بهتری ارائه گردد. همچنین در انجام محاسبات تا جای ممکن ساده سازی شده تا محاسبات سریع تر انجام شود.

بهینه سازی	اندازه کامپایل	میزان کاهش	پیشرفت
Optimization level -balance	15512	4384	28% کوچکتر
Link-Time Optimization	11128	232	2% کوچکتر
Split load store multiple	10896	0	بدون تغییر
MicroLib	10896	5924	38% کوچکتر
اندازه نهایی	4972	میزان پیشرفت	68% کوچکتر